Symbolic and Numerical Computation for Artificial Intelligence

edited by

Bruce Randall Donald
Department of Computer Science
Cornell University, USA

Deepak Kapur
Department of Computer Science
State University of New York, USA

Joseph L. Mundy
AI Laboratory
GE Corporate R&D, Schenectady, USA

Academic Press
Harcourt Brace Jovanovich, Publishers
London San Diego New York
Boston Sydney Tokyo Toronto

ACADEMIC PRESS LIMITED
24–28 Oval Road
London NW1

US edition published by
ACADEMIC PRESS INC.
San Diego, CA 92101

This book is printed on acid-free paper

A catalogue record for this book is available from the British Library

ISBN 0-12-220535-9

Printed and Bound in Great Britain by
The University Press, Cambridge

Machine Vision and Design

Chapter 5

Elimination Theory and Computer Vision: Recognition and Positioning of Curved 3D Objects from Range, Intensity, or Contours

Jean Ponce[†]

Department of Computer Science and Beckman Institute

University of Illinois

Urbana, IL 61801

David J. Kriegman

Department of Electrical Engineering and Center for Systems Science

Yale University

New Haven, CT 06520

An approach is presented for explicitly relating image observables to models of curved three-dimensional objects. This relationship is used for object recognition and positioning. Object models consist of collections of parametric surface patches. The image observables considered are raw range data, surface normals and Gaussian curvature, raw image contours, contour orientation and curvature, raw image intensity, and intensity gradient. Elimination theory provides a method for constructing an implicit equation that relates these observables to the three-dimensional position and orientation of object models. Determining the unknown pose parameters is reduced to a fitting problem between the implicit equation and the observed data points. Once the pose of candidate object models has been determined, recognition is achieved by computing the distance between the actual data points and the surface defined by the observables' equation. The proposed approach has been implemented and successfully tested on real images.

1. Introduction

Much progress has been made recently in three-dimensional object recognition from range (Bolles *et al.*, 1984; Besl and Jain, 1985; Hebert and Kanade, 1985; Faugeras and Hebert, 1986; Grimson and Lozano-Pérez, 1987; Fan *et al.*, 1988; Ikeuchi and Kanade,

[†] This research was funded in part by the UIUC Campus Research Board and by the National Science Foundation under Grants IRI-9015749 and DDM-9112458.

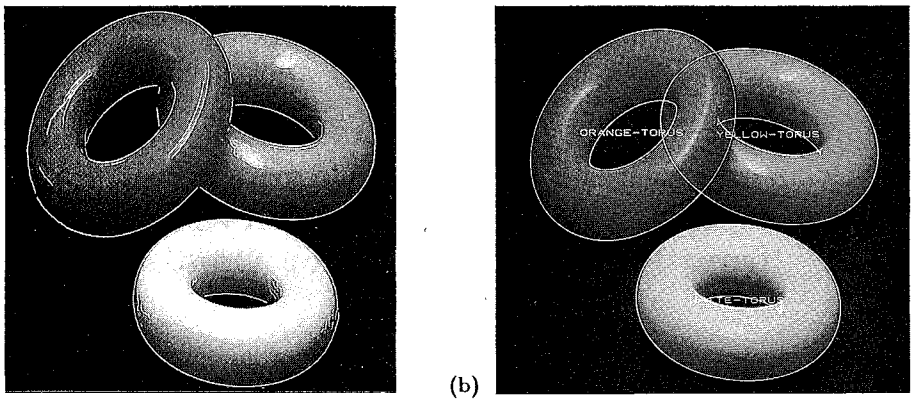


Figure 1. An example of recognition and positioning: (a) A real image of three plastic rings, with its Canny edges overlaid. (b) The recognized models are overlaid in the 3D position and orientation found by the algorithm described in section 3.

1988) and intensity (Horaud, 1987; Huttenlocher and Ullman, 1987; Lowe, 1987; Thompson and Mundy, 1987; Basri and Ullman, 1988) images. Typically, recognition is based on matching model and image features which have the same dimensionality: volumes are matched with volumes, surfaces with surfaces, curves with curves, and points with points. However, 3D model line segments may also be matched with 2D image segments (Horaud, 1987; Huttenlocher and Ullman, 1987; Lowe, 1987). The complexity of matching is at worst exponential in the number of features, but it is usually kept manageable by the use of so-called "rigidity constraints" (Faugeras and Hebert, 1986; Grimson, 1990), or "viewpoint consistency constraints" (Lowe, 1987), that simply express the fact that all model features are mapped into scene features through a single geometric transformation.

This approach has been quite successful when the objects to be recognized are polyhedra, but much less so when these objects have curved surfaces. In the latter case, most research has focused on segmenting images into surface or volume elements that can be directly compared to the surface and volume descriptions that comprise the object models. This includes recovery of quadrics (Faugeras and Hebert, 1986; Taubin, 1990), superquadrics (Pentland, 1986; Bajcsy and Solina, 1987; Gross and Boult, 1988), and generalized cylinders (Binford, 1971; Brooks, 1981; Ponce *et al.*, 1989) from range or intensity data. Little emphasis has been put on actual recognition, with the notable exception of Acronym (Brooks, 1981); see also promising new work based on invariant theory (Forsyth *et al.*, 1990; Taubin and Cooper, 1990).

This type of segmentation is expensive. For example, fitting a superquadric surface to a range image requires a non-linear minimization with respect to *fourteen* parameters (Bajcsy and Solina, 1987). Worse, in intensity images of curved objects, observable image features may *not* be the projection of model features. For example, the image contours of a polyhedron are the projection of polyhedral edges, but the contours of a smooth surface are the projection of perfectly regular points where the viewing direction happens to be tangent to the surface. In this paper, we propose to bypass the construction of an intermediate curve, surface, or volume representation from the image before matching. Instead, we match directly pointwise image observables to three-dimensional surface models.

Our models consist of collections of rational parametric surface patches. The reason for the choice of this representation is twofold. First and foremost, it allows us to represent geometric constraints by polynomial equations which, as shown below, are the key to our approach. Second, from a more pragmatic point of view, this is how objects are represented in most computer-aided design (CAD) systems. Databases of CAD models are now available for a wide range of manufactured objects; opportunities are better than ever for computer vision systems to exploit these models. As shown in Kriegman and Ponce (1990b), rational parametric patches also subsume nearly all representations used in computer vision, such as planes, quadrics, superquadrics, and generalized cylinders.

We consider a variety of pointwise observables, namely raw range data, surface normal orientation and Gaussian curvature, raw intensity data and intensity gradient, raw contour data and contour orientation and curvature. Elimination theory (Salmon, 1966; Dixon, 1908; Macaulay, 1916) provides an off-line method for constructing an implicit equation for each object model that relates these observables to the position and orientation of the object. Determining the pose parameters is reduced to a fitting problem between the observables' equation and the observed data points. Once the pose of candidate object models has been determined, recognition is achieved by computing the distance between the actual data points and the surface defined by the observables' equation.

The pose determination method has been implemented for the simple case of solids of revolution and experiments have been performed with synthetic and real range, intensity, and contour data (see figures 2 to 8 and Hoogs, 1991; Ponce *et al.*, 1991). Recognition from raw contour data has also been implemented (see figures 1,10,11 and Kriegman and Ponce, 1990b; Kriegman, 1989).

The rest of the paper is organized as follows. In section 2, elementary notions of elimination theory are presented, and rational parametric patches are defined. Section 3 introduces our approach to pose determination. This approach is applied to range data in section 4, to intensity data in section 5, and to contour data in section 6. A simple recognition algorithm is described in section 7. The current implementation is described in section 8 along with experimental results. Section 9 discusses future research. Bezout and Dixon resultants are presented in appendix.

This chapter is a synthesis of two papers detailing respectively our recognition experiments with contour data (Kriegman and Ponce, 1990b) and our positioning experiments with range, intensity, and contour data (Ponce *et al.*, 1991). The interested reader will find many implementation details in Kriegman (1989) and Hoogs (1991).

2. Elimination Theory and Rational Parametric Patches

2.1. Elimination Theory

Elimination theory (Salmon, 1966; Dixon, 1908; Macaulay, 1916) is a classical branch of mathematics which has been "rediscovered" in the context of computer graphics (Kajiya, 1982), computer-aided geometric design (Sederberg *et al.*, 1984; Goldman and Sederberg, 1985), robot kinematics (Buchberger, 1987; Raghavan and Roth, 1989), robot motion

planning (Schwartz and Sharir, 1987; Canny, 1988) and, recently, computer vision (Cyrluk *et al.*, 1988; Jerian and Jain, 1990; Kriegman and Ponce, 1990b).

The key idea of elimination theory is that a system of polynomials has common roots if and only if a single polynomial, called the *resultant* of the system of polynomials, vanishes. The original variables do not appear in the resultant, hence the name of elimination theory. The resultant itself is a polynomial in the coefficients of the original polynomials.

The simplest example of elimination is given by a well-known theorem of linear algebra: a necessary and sufficient condition for a square system of homogeneous linear equations to admit a nontrivial solution is that the determinant of this system vanishes.

The general case can be written as follows. Consider the following system of n polynomial equations in $n-1$ unknowns:

$$\begin{cases} P_1(x_1, \ldots, x_{n-1}) = 0, \\ \ldots \\ P_n(x_1, \ldots, x_{n-1}) = 0. \end{cases} \tag{2.1}$$

A necessary and sufficient condition for this system to admit a non-empty set of solutions is that:

$$R(P_1, \ldots, P_n) = 0, \tag{2.2}$$

where R is a polynomial in the coefficients of the P_i's. Note that these coefficients need not be numerical constants, but may contain further variables. On the other hand, the original variables x_i do not appear in R. The polynomial R is called the resultant of the P_i's, obtained by eliminating the variables x_1 to x_{n-1}.

Sylvester and Bezout resultants (Salmon, 1966) are available in most computer algebra systems, e.g. MACSYMA, MAPLE, MATHEMATICA, REDUCE. They can be readily used to eliminate one variable between two polynomials. Dixon's method (Dixon, 1908) can be used for eliminating two variables among three equations. These are sufficient for our purposes (see appendix). Elimination of $n-1$ variables between n polynomials can be achieved by eliminating these variables one by one with Sylvester or Bezout resultants, or by using other elimination methods (Salmon, 1966; Dixon, 1908; Macaulay, 1916; Buchberger, 1987; Canny, 1988).

2.2. RATIONAL PARAMETRIC PATCHES

In the rest of the paper, objects are modeled by rational parametric patches, whose Cartesian coordinates are ratios of bivariate polynomials, i.e.

$$\mathbf{x}(s,t) = \frac{1}{\sum_{i,j} w_{ij} s^i t^j} \sum_{i,j} s^i t^j \mathbf{x}_{ij}, \quad (s,t) \in I \times J, \tag{2.3}$$

where the coefficients w_{ij} are scalars, the coefficients $\mathbf{x}_{ij}$ are vectors, and I, J are intervals of $\mathbb{R}$. Bicubic patches are the most prominent type of surfaces in computer-aided design (Sederberg *et al.*, 1984; Farouki, 1987). They are given by the above equation with a constant denominator and a maximum total degree of 3 for s and t in the numerator.

As shown in Sederberg *et al.* (1984) and Goldman and Sederberg (1985), computing the implicit equation of a rational parametric patch is, in theory at least, a simple ex-

ercise in elimination. Equation (2.3) can be decomposed into three rational equations in the Cartesian coordinates x, y, z of the patch and its parametric coordinates s, t. As shown in Sederberg *et al.* (1984), Goldman and Sederberg (1985) and section 4, the implicit equation $F(x, y, z) = 0$ is obtained by clearing the denominators and eliminating s and t among the three equations. An implicit polynomial equation defining the intersection curve of two patches can be constructed by substituting the Cartesian coordinates $\mathbf{x}_1(s_1, t_1)$ of one patch into the implicit equation F_2 of the other patch. This equation is simply obtained by clearing the denominator of $F_2(\mathbf{x}_1(s_1, t_1)) = 0$, and it implicitly defines the curve in the parameter space of $\mathbf{x}_1$.

3. Relating Image Observables to Object Models

3.1. Principle of the method

For each type of data, we are going to derive a system of three equations of the form:

$$\begin{cases} f_1(s, t, \mathbf{O}, \mathbf{P}) = 0, \\ f_2(s, t, \mathbf{O}, \mathbf{P}) = 0, \\ f_3(s, t, \mathbf{O}, \mathbf{P}) = 0, \end{cases} \tag{3.1}$$

where f_1, f_2, and f_3 are rational functions in the parametric coordinates s and t, $\mathbf{O}$ is a vector of image observables, and $\mathbf{P}$ is a vector of viewing parameters mapping the world coordinate system onto the image coordinate system.

By clearing the appropriate denominators if necessary, these three equations can be transformed into a system of polynomial equations in s and t. Finally, by eliminating these two variables, we obtain a single equation

$$F(\mathbf{O}, \mathbf{P}) = 0 \tag{3.2}$$

that relates the image observables $\mathbf{O}$ to the viewing parameters $\mathbf{P}$.

Given a set of measurements $\mathbf{O}_i$, $i = 1, .., n$, recovering $\mathbf{P}$ is reduced to the (non-linear) least-squares minimization of:

$$\sum_{i=1}^{n} F^2(\mathbf{O}_i, \mathbf{P}) \tag{3.3}$$

with respect to $\mathbf{P}$.

3.2. Notations

Consider a parametric patch $\mathbf{x}$, defined by (2.3) in the coordinate system $(\mathbf{o}, \mathbf{i}, \mathbf{j}, \mathbf{k})$. The normal to the patch is $\mathbf{n} = \mathbf{x}_s \times \mathbf{x}_t$, where $\mathbf{x}_s$ (resp. $\mathbf{x}_t$) is the partial derivative of $\mathbf{x}$ with respect to s (resp. t), and "$\times$" denotes the cross-product operator. The unit normal is $\mathbf{N} = \frac{1}{|\mathbf{n}|}\mathbf{n}$. The coefficients of the first (resp. second) fundamental form in the basis $(\mathbf{x}_s, \mathbf{x}_t)$ are denoted E, F, G (resp. e, f, g). The second fundamental form itself is denoted II.

We consider an image coordinate system $(\mathbf{c}, \mathbf{v}, \mathbf{w}, \mathbf{u})$ where $\mathbf{c}$ is the camera origin.†

† For simplicity, we assume here orthographic projection, but the methods proposed in this paper

The unit vector $\mathbf{v}$ is the viewing direction, with spherical coordinates (α, β) in the coordinate system $(\mathbf{i}, \mathbf{j}, \mathbf{k})$, and $(\mathbf{w}, \mathbf{u})$ is an orthonormal basis for the image plane. The remaining degree of freedom of the rotation mapping the coordinate system $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ into the coordinate system $(\mathbf{w}, \mathbf{u}, \mathbf{v})$ can be represented by some rotation angle γ in the image plane.

Let x, y denote the image coordinates of a point, and z denote its depth. These coordinates are given by:

$$\begin{cases} x = (\mathbf{x} - \mathbf{c}) \cdot \mathbf{w}, \\ y = (\mathbf{x} - \mathbf{c}) \cdot \mathbf{u}, \\ z = (\mathbf{x} - \mathbf{c}) \cdot \mathbf{v}. \end{cases} \tag{3.4}$$

Let $x_0 = -\mathbf{c} \cdot \mathbf{w}$, $y_0 = -\mathbf{c} \cdot \mathbf{u}$, $z_0 = -\mathbf{c} \cdot \mathbf{v}$, so that (x_0, y_0, z_0) are the coordinates of $\mathbf{o}$ in the camera-centered coordinate system $(\mathbf{c}, \mathbf{w}, \mathbf{u}, \mathbf{v})$. The above equations become:

$$\begin{cases} x = \mathbf{x} \cdot \mathbf{w} + x_0, \\ y = \mathbf{x} \cdot \mathbf{u} + y_0, \\ z = \mathbf{x} \cdot \mathbf{v} + z_0. \end{cases} \tag{3.5}$$

The vector of viewing parameters is in general $\mathbf{P} = (\alpha, \beta, \gamma, x_0, y_0, z_0)$.

4. Range Data

In this section, we suppose that range information is available as a dense range image $z(x, y)$ acquired by a laser rangefinder, for example. The proposed approach readily applies to sparse range data obtained, for example, from a tactile sensor (Salisbury, 1984; Fearing, 1987). For approaches to tactile data interpretation, see Fearing (1987), Gunnarsson and Prinz (1987) and Allen and Michelman (1990).

Figures 2 and 3 show examples of positioning from range data (see section 8 for details).

4.1. Raw data

The raw data of a range image can be described by (3.5). All three equations are rational in the parametric coordinates s and t, and by eliminating these two variables, a single equation is obtained:

$$F(x, y, z, \alpha, \beta, \gamma, x_0, y_0, z_0) = 0. \tag{4.1}$$

Here, the observables are the image coordinates and the range at each point, i.e. $\mathbf{O} = (x, y, z)$. The vector of viewing parameters is $\mathbf{P} = (\alpha, \beta, \gamma, x_0, y_0, z_0)$.

It should be noted that we have just "rediscovered" the implicitization of a rational patch (Sederberg *et al.*, 1984; Goldman and Sederberg, 1985). F is simply the implicit equation of the patch $\mathbf{x}$, parameterized by its position and orientation. It should also be noted that the idea of using the implicit equation of algebraic surfaces (namely quadrics) for object positioning from tactile data was proposed in Gunnarsson and Prinz (1987).

readily extend to scaled orthography and perspective projection (Kriegman and Ponce, 1990b; Hoogs, 1991).

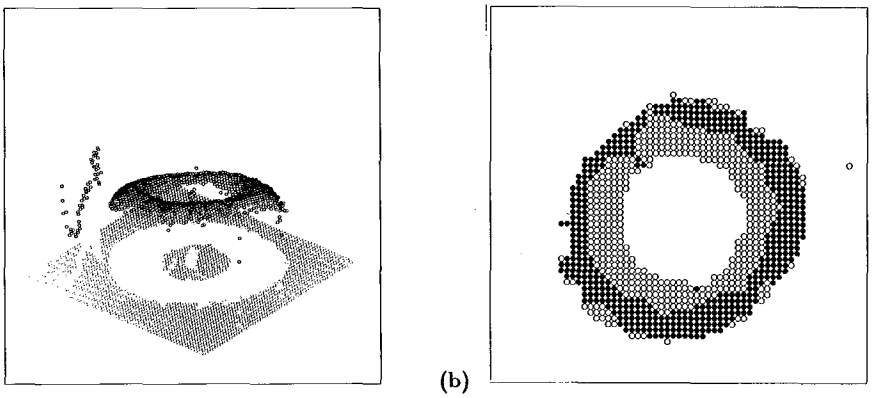


Figure 2. Range data: (a) A perspective view of a range image. (b) The elliptic (solid discs) and hyperbolic (circles) points of the torus in this image.

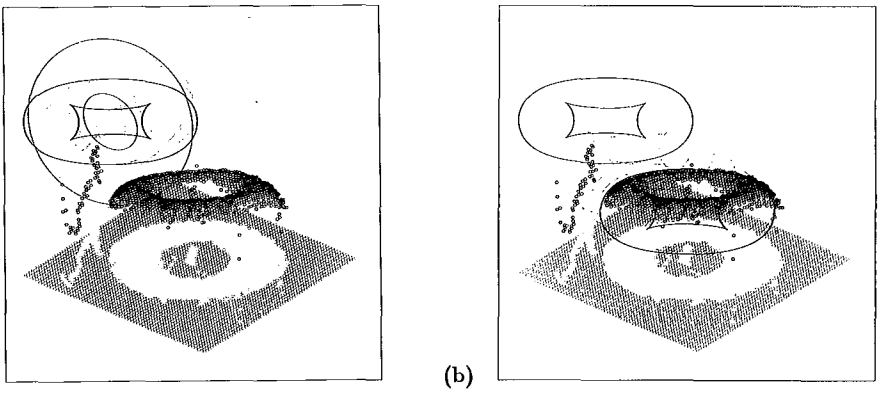


Figure 3. Positioning from range data: (a) Orientation estimation. (b) Translation estimation.

4.2. Orientation and Gaussian curvature

More interestingly, the translation components of the viewing transformation can be bypassed by using translation-independent observables such as surface normal and Gaussian curvature, whose reliable computation has been demonstrated many times in the past (see, for example, Besl and Jain, 1985; Brady *et al.*, 1985; Fan *et al.*, 1987; Besl and Jain, 1988).

Let θ and ϕ be the spherical coordinates of the *observed* unit surface normal $\mathbf{N}$ in the coordinate system $(\mathbf{w}, \mathbf{u}, \mathbf{v})$. We have:

$$\mathbf{N}(\theta, \phi) \times \mathbf{n}(s, t) = \mathbf{0}, \tag{4.2}$$

where the coordinates of $\mathbf{n}$ are in the basis $(\mathbf{w}, \mathbf{u}, \mathbf{v})$. Also, note that only two of the three scalar equations given by (4.2) are independent.

The Gaussian curvature is given by do Carmo (1976, p. 155):

$$K = \frac{eg - f^2}{EG - F^2}, \tag{4.3}$$

where e, f, g, E, F, and G are also functions of the parametric coordinates s and t.

Equations (4.2) and (4.3) are rational in s, t, K, and in trigonometric functions of $\theta, \phi, \alpha, \beta$, and γ. By eliminating s and t, a single equation is obtained:

$$F(K, \theta, \phi, \alpha, \beta, \gamma) = 0. \tag{4.4}$$

Here, the vector of observables is $\mathbf{O} = (K, \theta, \phi)$, and the vector of viewing parameters is $\mathbf{P} = (\alpha, \beta, \gamma)$. It is worth noting that we have just derived the implicit equation (parameterized by orientation) of Extended Gaussian Images (Horn, 1984). Once α, β, and γ have been found, a second three-parameter minimization can be used to determine x_0, y_0, z_0. The optimal values for α, β, γ are substituted in (4.1); they are kept constant, and the minimization is performed with respect to the unknowns x_0, y_0, z_0 only.

5. Image Intensity

We now consider a matte surface patch observed under orthographic projection. We assume unit surface albedo and a single distant light source with direction $\mathbf{l}$, and denote by θ, ϕ the (unknown) spherical coordinates of the light source in the camera-centered coordinate system $(\mathbf{w}, \mathbf{u}, \mathbf{v})$.

Figure 4 shows examples of positioning from intensity data (see section 8 for details).

5.1. Raw data

As shown by (3.5), under orthography, the image coordinates x, y of a point are given by:

$$\begin{cases} x = \mathbf{x} \cdot \mathbf{w} + x_0, \\ y = \mathbf{x} \cdot \mathbf{u} + y_0. \end{cases} \tag{5.1}$$

The intensity is given by:

$$I = \mathbf{N} \cdot \mathbf{l}. \tag{5.2}$$

The system of three equations formed by (5.1) and squaring (5.2) is rational in s and t. Eliminating these two variables, a single equation is obtained:

$$F(x, y, I, \alpha, \beta, \gamma, x_0, y_0, \theta, \phi) = 0. \tag{5.3}$$

Here, there are three observables, $\mathbf{O} = (x, y, I)$, and seven viewing parameters (including light source direction), $\mathbf{P} = (\alpha, \beta, \gamma, x_0, y_0, \theta, \phi)$. It should be noted that F is simply the implicit equation of the image intensity surface $I(x, y)$, parameterized by the position and orientation of the patch $\mathbf{x}$ and the direction of the light source $\mathbf{l}$. Note that any implementation of this approach will have to incorporate at least one additional parameter accounting for albedo and light source intensity (i.e. the parameter a in $I = a\mathbf{N} \cdot \mathbf{l}$).

5.2. Gradient

The implicit equation of the image intensity surface depends on the translation parameters of the viewing transformation. By measuring intensity gradient as well as intensity,

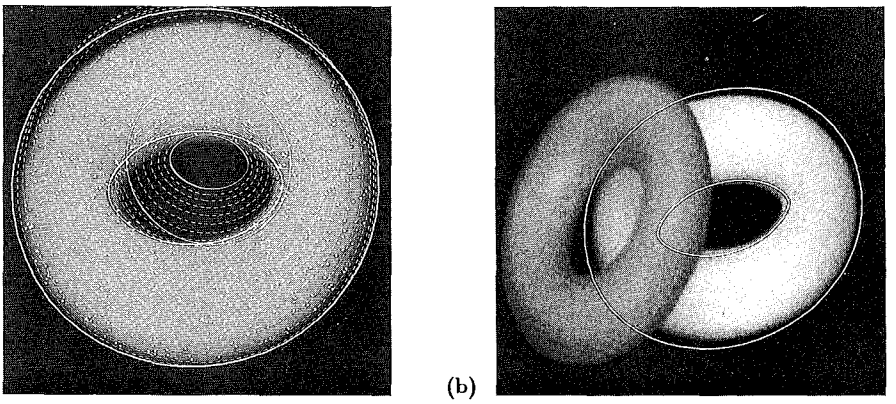


Figure 4. Positioning from intensity data: (a) A simple case without occlusion. The data points are shown as white dots. (b) Another example with occlusion.

an alternative translation-independent relationship is obtained. The gradient of the intensity with respect to the parametric coordinates is given by:

$$\begin{cases} I_s = \mathbf{N}_s \cdot \mathbf{l}, \\ I_t = \mathbf{N}_t \cdot \mathbf{l}. \end{cases} \tag{5.4}$$

The image coordinates are given as before by (5.1). Differentiating these equations with respect to s and t, we obtain:

$$\begin{cases} \dfrac{\partial x}{\partial s} = \mathbf{x}_s \cdot \mathbf{w}, & \dfrac{\partial x}{\partial t} = \mathbf{x}_t \cdot \mathbf{w}, \\ \dfrac{\partial y}{\partial s} = \mathbf{x}_s \cdot \mathbf{u}, & \dfrac{\partial y}{\partial t} = \mathbf{x}_t \cdot \mathbf{u}. \end{cases} \tag{5.5}$$

The image gradient can now be written as:

$$\begin{cases} I_x = I_s \dfrac{\partial s}{\partial x} + I_t \dfrac{\partial t}{\partial x}, \\ I_y = I_s \dfrac{\partial s}{\partial y} + I_t \dfrac{\partial t}{\partial y}. \end{cases} \tag{5.6}$$

So finally, the following system of equations is obtained:

$$\begin{cases} I = \mathbf{N} \cdot \mathbf{l}, \\ I_x = \dfrac{\mathbf{N_s} \cdot \mathbf{l}}{\mathbf{x}_s \cdot \mathbf{w}} + \dfrac{\mathbf{N_t} \cdot \mathbf{l}}{\mathbf{x}_t \cdot \mathbf{w}}, \\ I_y = \dfrac{\mathbf{N_s} \cdot \mathbf{l}}{\mathbf{x}_s \cdot \mathbf{u}} + \dfrac{\mathbf{N_t} \cdot \mathbf{l}}{\mathbf{x}_t \cdot \mathbf{u}}. \end{cases} \tag{5.7}$$

These three equations are (after squaring) rational in s and t, and a single equation is obtained upon eliminating these variables:

$$F(I, I_x, I_y, \alpha, \beta, \gamma, \theta, \phi) = 0. \tag{5.8}$$

Here, the observables are $\mathbf{O} = (I, I_x, I_y)$. Once again, the vector of viewing parameters $\mathbf{P} = (\alpha, \beta, \gamma, \theta, \phi)$ includes the light source direction but, this time, it does not depend on the translation parameters x_0, y_0. As before, these parameters can be computed after a second minimization.

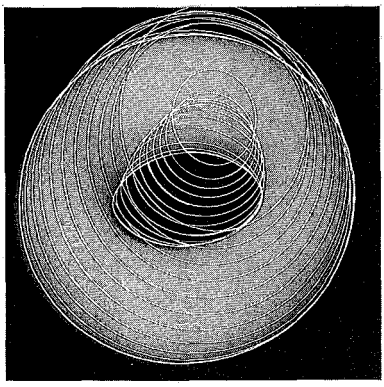


Figure 5. Positioning from raw contour data.

6. Image Contours

For matte surfaces, image contours are zero order intensity discontinuities, formed by surface normal discontinuities (creases, edges, or corners), depth discontinuities (occluding contours, or limbs), reflectivity discontinuities (pigmentation or material changes), or lighting discontinuities (shadows). Below, we show how to obtain implicit equations for the image contours formed by the projection of occluding contours, where the viewing direction is tangent to the surface. Implicit equations for the contours corresponding to surface normal and reflectivity discontinuities can be obtained in a similar way (Kriegman and Ponce, 1990b). The extension to shadows requires the introduction of parameters describing the light source direction as in the previous section.

Figures 5,7 and 8 show examples of positioning from contour data (see section 8 for details).

6.1. Raw data

Occluding contours are characterized by the viewing direction being tangent to the surface, or:

$$\mathbf{n} \cdot \mathbf{v} = 0. \tag{6.1}$$

As seen before, (5.1) gives the image coordinates x, y of a point under orthographic projection. Equations (5.1) and (6.1) are rational in s and t, and by eliminating these two variables, we obtain an expression of the form:

$$F(x, y, \alpha, \beta, \gamma, x_0, y_0) = 0, \tag{6.2}$$

which is the implicit equation of the image contours.

Here, the vector of observables is $\mathbf{O} = (x, y)$, and the vector of viewing parameters is $\mathbf{P} = (\alpha, \beta, \gamma, x_0, y_0)$. It should be noted that F is simply the implicit equation of the image contours of the patch $\mathbf{x}$, parameterized by the position and orientation of this patch. Details for this case can be found in Kriegman and Ponce (1990b).

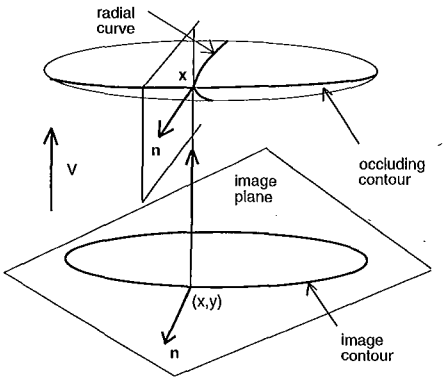


Figure 6. The geometry of Koenderink's theorem. The radial curve is the intersection of the patch and the plane that contains **n** and **v**. The radial curvature is its curvature at **x**.

6.2. Orientation and curvature

Under orthographic projection, the normal to the image contour is aligned with the surface's normal. It follows that at a contour point, the angle θ between the **w** axis and the normal to the contour satisfies the constraint:

$$\mathbf{N} \cdot \mathbf{w} = \cos\theta. \tag{6.3}$$

As shown by Koenderink (1984, 1986), the contour curvature κ and the Gaussian curvature K are related by:

$$K = \kappa\kappa_r, \tag{6.4}$$

where κ_r is the "radial" curvature, i.e. the normal curvature of the surface in the viewing direction (figure 6).†

It is easy to show that the coordinates (ds, dt, dn) of the vector **v** in the coordinate system $(\mathbf{x}_s, \mathbf{x}_t, \mathbf{n})$ are given by:

$$\begin{cases} ds = \dfrac{(\mathbf{x}_s \cdot \mathbf{v})G - (\mathbf{x}_t \cdot \mathbf{v})F}{EG - F^2}, \\ dt = \dfrac{E(\mathbf{x}_t \cdot \mathbf{v}) - F(\mathbf{x}_s \cdot \mathbf{v})}{EG - F^2}, \\ dn = \dfrac{\mathbf{n} \cdot \mathbf{v}}{EG - F^2}. \end{cases} \tag{6.5}$$

From (6.1), $dn = 0$ at an occluding contour point. The normal curvature in the direction of the unit vector **v** is equal to the value of the second fundamental form in this direction (do Carmo, 1976, pp. 142,154), i.e.:

$$\kappa_r = II(\mathbf{v}) = eds^2 + 2fdsdt + gdt^2. \tag{6.6}$$

So finally:

$$\kappa = \frac{1}{eds^2 + 2fdsdt + gdt^2} \frac{eg - f^2}{EG - F^2}. \tag{6.7}$$

† Similar results still hold under spherical (Koenderink, 1984) and pinhole perspective projection (Vaillant, 1990).

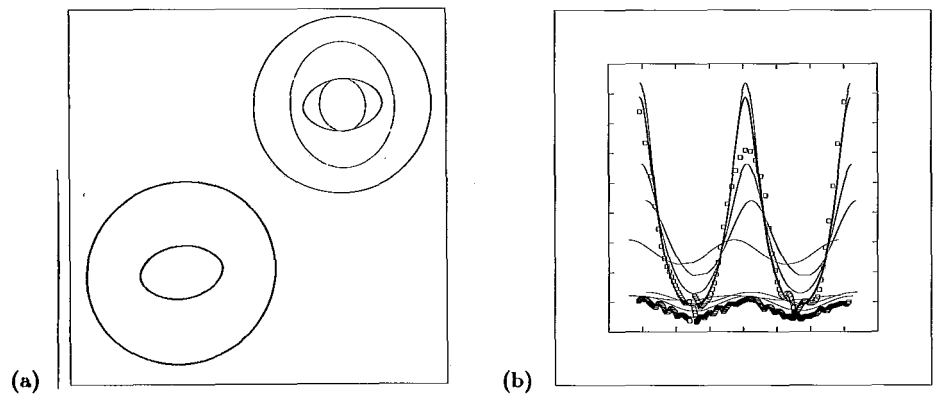


Figure 7. Orientation estimation from contour orientation and curvature: (a) Orientation estimation in image space. (b) Iterations in orientation-curvature space.

Equations (6.1), (6.3) and (6.7) are rational in s and t, and after eliminating these two variables, we obtain:

$$F(\kappa, \theta, \alpha, \beta, \gamma) = 0. \tag{6.8}$$

Here, the observables are $\mathbf{O} = (\kappa, \theta)$, and the vector of viewing parameters is once again $\mathbf{P} = (\alpha, \beta, \gamma)$ which no longer includes the translation parameters x_0, y_0. Once the optimal rotation parameters have been determined, the translation parameters can then be computed through a second minimization.

7. A Simple Recognition Algorithm

We now present a simple recognition algorithm:

0. For the appropriate sensing modality, compute off-line the functions F_j, $j = 1, .., m$ corresponding to each element in a database of models as described above. On-line, for a set of measured data points $\mathbf{O}_i$, $i = 1, .., n$, do:

1. For each model j, minimize the squared error:

$$E_j = \sum_{i=1}^{n} F_j^2(\mathbf{O}_i, \mathbf{P}) \tag{7.1}$$

with respect to the components of $\mathbf{P}$.

2. For each model j, let $\mathbf{P}_j$ be the optimal parameters computed at the previous step, compute the cumulative distance $D_j = \sum_{i=1}^{n} d_{ij}$ between the data points $\mathbf{O}_i$ and the surface defined by F_j and $\mathbf{P}_j$ as described below.

3. Label the data points with the model j_0 that minimizes D_j.

Notice that the residual E_j is not used to distinguish between different models because it is *not* a distance measure. Instead, we use the actual distance between the data points and the surface defined by F_j. We now present two algorithms for computing this distance.

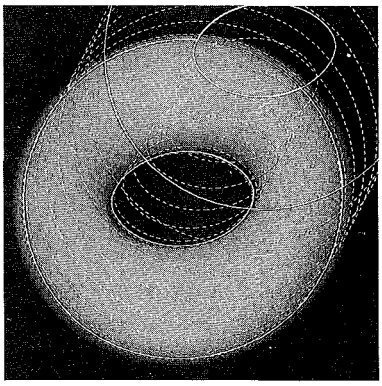


Figure 8. Estimation of translation from contours with fixed orientation.

7.1. Exact distance function

The distance between a data point $\mathbf{O}_i$ and the surface described by F_j for parameter values $\mathbf{P}_j$ is defined as the minimum of the distance between this point and the surface. This minimum is reached at a point $\mathbf{O}$ on the surface where the surface normal and the line joining $\mathbf{O}$ and $\mathbf{O}_i$ are aligned.†

The distance is therefore given by $d_{ij}^2 = |\mathbf{O} - \mathbf{O}_i|^2$, where the point $\mathbf{O}$ is the solution of:

$$\begin{cases} F_j(\mathbf{O}, \mathbf{P}_j) = 0, \\ (\mathbf{O} - \mathbf{O}_i) \times \nabla F_j(\mathbf{O}, \mathbf{P}_j) = 0, \end{cases} \tag{7.2}$$

where "$\times$" denotes the operator that associates the $p-1$ vector $(x_1 y_2 - x_2 y_1, .., x_{p-1} y_p - x_p y_{p-1})$ to the two vectors $(x_1, .., x_p)$ and $(y_1, .., y_p)$. (An abuse of the cross-product notation. Clearly, $\mathbf{x}_1 \times \mathbf{x}_2 = \mathbf{0}$ if and only if $\mathbf{x}_1$ and $\mathbf{x}_2$ are aligned.)

This is a system of p equations in p unknowns, where p is the number of observables (the dimension of $\mathbf{O}$). As seen in the previous sections, each F_j is rational in the observables, so by clearing the appropriate denominators, we obtain a system of p polynomial equations in p variables.

To solve this system and compute the distance d_{ij}, we can use one of two methods. First, by adding to this system the equation $d_{ij}^2 = |\mathbf{O} - \mathbf{O}_i|^2$, we can eliminate the p observables among $p+1$ equations to obtain a single equation:

$$D(d_{ij}, \mathbf{O}_i, \mathbf{P}_j) = 0, \tag{7.3}$$

where D is a polynomial in the distance d_{ij}, parameterized by the components of $\mathbf{O}_i$ and $\mathbf{P}_j$. For a given set of parameters $\mathbf{P}_j$, the distance d_{ij} is the minimum positive root of this polynomial, and it can be found by some numerical root-finding algorithm (Press *et al.*, 1986). For a set of points found by an edge detector, figure 9 shows the nearest points on the image contours of a torus model. Also, note that since d_{ij} is given by an implicit equation, it is possible to compute its derivatives with respect to the viewing parameters; this is useful for numerical minimization.

† Actually, a minimum can also be reached at a singular point of the F_j surface where the gradient vanishes, but this case is also captured by (7.2).

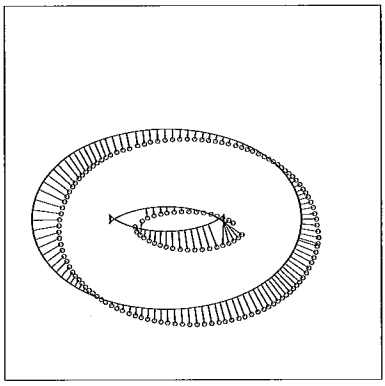


Figure 9. The exact distance between contour points and some theoretical contour.

An alternative solution is to solve directly (7.2) by using a global numerical method such as homotopy continuation (Morgan, 1987). In principle, the exact distance computed by either method can be used instead of F for pose estimation. In practice, both methods may prove too inefficient to be used directly during the minimization since for each iteration, the roots of D must be found for all data points. However, D can still be used to compute the exact distance between the data points and the optimal surface after convergence.

7.2. Approximate distance function

We now propose a method to compute an *approximate* value of the distance after convergence. As previously noted, the minimum distance between $\mathbf{O}_i$ and the surface F_j is reached at a surface point $\mathbf{O}$ where the surface normal is aligned with the line joining $\mathbf{O}_i$ and $\mathbf{O}$. Suppose that we have an estimate $\mathbf{N}_i$ of the surface normal at the point $\mathbf{O}_i$. The minimum distance is reached at a point $\mathbf{O} = \mathbf{O}_i + \lambda \mathbf{N}_i$ such that $F_j(\mathbf{O}, \mathbf{P}_j) = 0$. This can be rewritten as an equation in λ:

$$F_j(\mathbf{O}_i + \lambda \mathbf{N}_i, \mathbf{P}_j) = 0. \tag{7.4}$$

Since a good initial estimate of λ is available from fitting (i.e. $\lambda = 0$), Newton-Raphson iterations can be used to find the first zero of this equation, and the distance is readily obtained.

For all three of our sensing modalities (range, intensity, contours), the normal to the surface corresponding to the raw data can be estimated easily through numerical differentiation: surface normal from range data, intensity gradient from intensity data, contour normal from contour data. If the data points are close to the actual surface, the computed normal is a good approximation of the actual normal. This suggests a refined recognition scheme. First, compute the orientation parameters using a translation-free minimization, then compute the translation corresponding to the optimal orientation. Finally, compute the distance by the above method.

It should also be noted that Taubin has proposed an alternative approximation to the distance function which can be used for either fitting or recognition with the pa-

rameterized implicit equations given in sections 4 to 6 (Taubin, 1990). However, the corresponding equation is of higher degree than the original implicit equation used here.

8. Implementation and Results

8.1. Implementation

To demonstrate the feasibility of our approach, we have implemented the pose determination algorithm presented in previous sections for a limited world made of various solids of revolution observed under weak perspective projection.

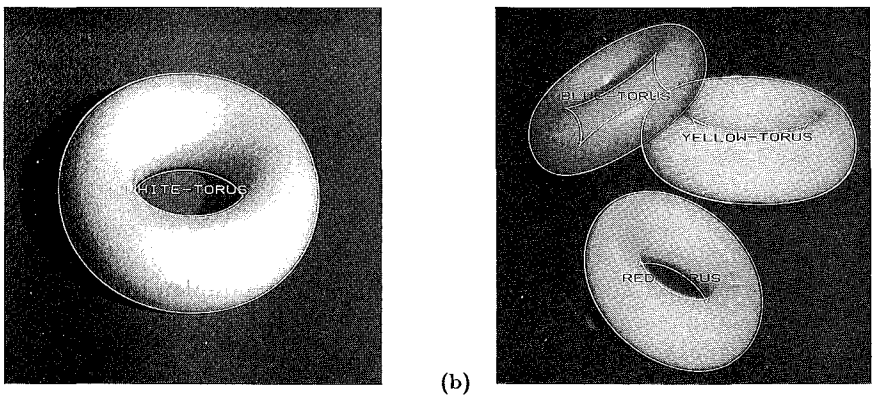
Why such a choice? Solids of revolution are obtained by rotating a planar generating curve about a straight line; they are at the same time simple enough (due to their symmetry, three-dimensional pose has only five independent parameters in this case, and the elimination of two variables can be replaced by the elimination of a single variable) and complicated enough (non-convex surfaces with non-planar occluding contours) to qualify as initial test objects. See Hoogs (1991) for a detailed account of the solid of revolution case.

Weak perspective (or scaled orthography) has been chosen because, unlike orthography, it models accurately most imaging conditions and, unlike perspective, it does not require camera calibration. In addition, the current implementation is limited to surface normal orientation and Gaussian curvature data, raw contour data, contour curvature and orientation data, and raw intensity data. Note that recognition has also been implemented, but only for the case of raw contour data (Kriegman and Ponce, 1990b).

Elimination is performed by the Reduce implementation of the resultant of two polynomials. Note that several other computer algebra systems are commercially available (e.g. MACSYMA, MAPLE, MATHEMATICA), and they all offer some version of the resultant of two polynomials (Sylvester or Bezout resultant), which is sufficient for the case of surfaces of revolution. Elimination of two variables as required in the case of general parametric patches can be achieved in two passes by using these resultants or directly by using some implementation of Dixon's resultant (Dixon, 1908) or general multivariate resultants (Canny, 1988).

Like many others (e.g. Bajcsy and Solina, 1987; Gross and Boult, 1988), we use the Levenberg-Marquardt algorithm (Press *et al.*, 1986) to solve the nonlinear least squares minimization of (3.3). Note that this algorithm only finds a local minimum and therefore needs a reasonable set of initial parameters. As shown in the examples of the next section, this has not been a problem in our case. Note also that when the rotation and translation parameters are estimated independently, a post-processing step could be added where these estimates are fed as initial parameters to the minimization of the raw data equation, and use a couple of iterations to refine these estimates simultaneously. We have not tried this refined method so far.

Finally, it should be noted that we do not claim to avoid segmentation: the points corresponding to a given surface must be identified before the minimization takes place. So far, this has been done by hand. As discussed in section 9, better, automated segmentation is left for future work. Our algorithms have been implemented in Common Lisp



(a) (b)

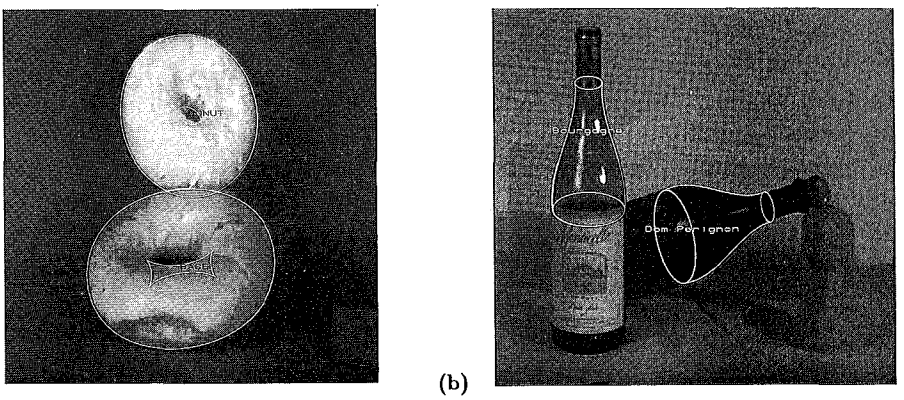
Figure 10. Recognition examples: (a) An isolated torus. (b) Three tori with occlusion.

and run on Symbolics Lisp Machines, SUN SparcStations, and Silicon Graphics Personal Irises. All positioning timings reported here are in user-time seconds on a SUN SparcStation 1 running Lucid Common Lisp. All recognition timings are in user-time seconds on a Symbolics 3600 Lisp Machine.

8.2. Results

Figures 2 to 8 demonstrate the implementation of positioning on a variety of plastic rings modeled as tori and different types of real data (see Hoogs, 1991, for more examples, including other solids of revolution and synthetic data). In these figures, the silhouette of the torus considered is overlaid on the original image. The initial pose is drawn in thin solid lines, the successive pose estimates are shown as light, dotted lines, and the estimate obtained at convergence is drawn in thick solid lines. In each case, the initial parameters of the minimization have been arbitrarily set, relatively far away from the nominal parameters. In our examples, convergence toward a global optimum has been observed, but such global convergence is not in general guaranteed by local optimization procedures like the Levenberg-Marquardt algorithm.

Figures 2 and 3 show our experiments with real range data, kindly provided by Dr. Martial Hebert from Carnegie-Mellon University. A perspective view of a range image of a torus is shown in figure 2(a). We have not tried to compute the pose of the torus directly from range data since other authors have already reported similar experiments (Gunnarsson and Prinz, 1987). Instead, we have concentrated on estimating the orientation parameters from surface orientation and Gaussian curvature. These observables were computed using the method described in Brady *et al.* (1985). Briefly, the range data is first smoothed using repeated local averaging (Burt, 1981), then first and second image derivatives are estimated through finite differences, and the normal and Gaussian curvature are computed using the formulas given earlier. Figure 2(b) gives a qualitative idea of the performance of this algorithm by showing points classified as elliptic or hyperbolic. Figure 3(a) shows the orientation computed by the program after 10 Levenberg-Marquardt iterations. There are 953 data points, and the minimization takes 9.5s. Once



(a) (b)

Figure 11. Breakfast and dinner: (a) A doughnut and a bagel. (b) Wine and Champagne bottles.

the orientation has been computed, the translation is computed from the raw range data, as shown in figure 3(b). The computing time in this case is 6.3s.

Figure 4 shows the results of experiments with real intensity data. Here, we have assumed that the light source is in the same direction as the observer, which reduces the number of estimated parameters from eight to six (two orientation parameters, two translations parameters, plus scale and albedo). In figure 4(a), the object has been separated from its background by thresholding. Note that this is the only image processing performed. In this experiment, the total computing time is 21.9s for 645 points (we have only used a (random) subset of the points that belong to the torus as data; these are shown as white dots in figure 4). We have not experimented with intensity gradient yet. Figure 4(b) shows another example with occlusion. In this case, the data points corresponding to the target torus have been selected by hand (actually by keeping only the data from the right half of the image).

Finally, we have experimented with contour data (figures 5,7 and 8). Figure 5 shows the results of an experiment using raw contours. All five pose parameters are estimated in this case. Here, the outline of the object (hence, the edge points) has been found by thresholding the image. In other experiments (Kriegman and Ponce, 1990b), we have used instead the Canny edge detector (Canny, 1986), hand-selecting the subset of edgels corresponding to the outline of the target object. Here, this has not been necessary because the torus is isolated and the contrast is quite good. The computing time for 1247 points is 219.8s. In our next set of experiments, we used contour orientation and curvature instead of raw contour data. Accurate estimates of these quantities were computed by using a contour approximation program kindly provided by Régis Vaillant from INRIA (Vaillant, 1990). Briefly, this program first computes a spline approximation of the contour using the Numerical Analysis Group (NAG) library of numerical routines, and then computes orientation and curvature analytically from the spline representation. Figure 7(a) shows the orientation and scale (three parameters) computed from these estimates by our program for the same image as before. Some edgels are rejected by the program, so that there are only 1205 data points, and the computing time is 11.0s. This compares favorably with the 219.8s required by the raw contour minimization on the same problem. Figure 7(b) shows the iterations in orientation/curvature space. Once the orientation and scale have been computed, these parameters can be substituted in the

raw contour equation, and figure 8 shows the corresponding two-parameter estimation of translation, which took 8.8 seconds using the same data points.

Finally, figures 1,10 and 11 show examples of recognition. The models used in our first set of experiments are five plastic rings that will be referred to by their colors: red, white, blue, orange, and yellow, even though they are really distinguished in our black and white images by the fact that the ratios of their major radius over their minor radius are all different. Figure 10(a) shows the result of recognition for an isolated torus. All five models are fit, and the correct model (white) is recognized. Figures 1 and 10(b) show more complicated examples, with three different tori and occlusion. Again, the correct tori are recognized. The total computing times on a Symbolics Lisp machine for these three examples are respectively 15, 33 and 32 seconds. As shown in Kriegman and Ponce (1990b), the average distance between the theoretical contours and the data points varies between 0.57 and 1.08 pixels for the recognized models.

Figure 11 shows two more examples. In figure 11(a), a bagel and a doughnut are recognized and positioned correctly, even though these objects are poorly approximated by tori. The computing time is 10 seconds in this case, and the average distance for the two objects is about 1.5 pixels. Finally, figure 11(b) shows a bottle of California Pinot Noir, with its typical "Bourgogne" shape, and a bottle of Dom Perignon Champagne. In both cases, only the doubly curved portion of the surface has been modeled. The scaling is modeled by a cubic curve. The recognition time in this case is 20 seconds, and the average distance is 0.43 and 0.46 pixels for the two recognized bottles.

9. Discussion and Future Work

Elimination theory has been used to relate explicitly pointwise image observables such as range, intensity, or contours, to the position and orientation of parametric patches. In turn, this relationship has been used in an implemented algorithm for locating solids of revolution either from surface orientation and Gaussian curvature, raw intensity, raw contours, or contour orientation and curvature. An algorithm for recognizing solids of revolution in raw contour data has also been presented.

Our immediate goal is to experiment with more general objects, using CAD models of piecewise-smooth objects made of collections of more general parametric surface patches (Kriegman and Ponce, 1991).

We are also investigating recognition and positioning of curved objects for which CAD models are *not* available. This entails discovering (learning) shape models from a sequence of images. Instead of precise object models made of (possibly many) parametric patches, we will use simplified shape descriptions in the form of low-degree implicit algebraic surfaces. The methods presented in this paper readily extend to these surfaces. Here, the equation $F(\mathbf{O}, \mathbf{P}) = 0$ has been used to recover the viewing parameters $\mathbf{P}$ when the shape of the observed object was known. Shape discovery is dual to shape positioning, and it can proceed from the same equation by assuming the transformation is known through calibration and solving instead for the shape parameters.

Segmentation is one of the most difficult problems in computer vision. In this chapter, we have avoided high-level segmentation of images into curves, surfaces, and volumes, but data points corresponding to a single surface have been selected by hand. Clearly,

methods will have to be developed for deciding automatically which image points should be matched with which model surfaces.

We have focused on positioning methods rather than recognition strategies. We are currently investigating strategies suited to our positioning approach, in the same way as interpretation trees have been used in conjunction with so-called "rigidity constraints" (Faugeras and Hebert, 1986; Grimson and Lozano-Pérez, 1987) or "viewpoint consistency constraints" (Lowe, 1987) in the case of polyhedra (see also Horaud, 1987; Huttenlocher and Ullman, 1987; Thompson and Mundy, 1987). These strategies are based on using aspect graphs (Koenderink and Van Doorn, 1979; Eggert and Bowyer, 1989; Sripradisvarakul and Jain, 1989; Kriegman and Ponce, 1990a; Ponce and Kriegman, 1990a) to represent qualitatively an object's image features such as contours and t-junctions, and taking into account the quantitative constraints imposed by these features (Ponce and Kriegman, 1990b; Seales and Dyer, 1991).

Acknowledgments

We would like to thank Anthony Hoogs for his participation to this work, Martial Hebert for supplying the range data used in these experiments and Régis Vaillant for providing his spline approximation program. We also thank Seth Hutchinson for numerous discussions and comments.

References

P. Allen and P. Michelman (1990), "Acquisition and interpretation of 3-D sensor data from touch", *IEEE Trans. Robotics and Automation*, **6**(4), 397–404.

R. Bajcsy and F. Solina (1987), "Three-dimensional object representation revisited", *Proc. Int. Conf. Comput. Vision*, London, UK, 231–240.

R. Basri and S. Ullman (1988), "The alignment of objects with smooth surfaces", *Proc. Int. Conf. Comput. Vision*, Tampa, FL, 482–488.

P.J. Besl and R.C. Jain (1985), "Three-dimensional object recognition", *ACM Comput. Surveys*, **17**(1), 75–145.

P.J. Besl and R.C. Jain (1988), "Segmentation through variable-order surface fitting", *IEEE Trans. Patt. Anal. Mach. Intell.*, **10**(2), 167–192.

T.O. Binford (1971), "Visual perception by computer", *Proc. IEEE Conf. Syst. Control*, Miami, FL.

R.C. Bolles, P. Horaud and M.J. Hannah (1984), "3DPO: a three-dimensional part orientation system", *Proc. Int. Symp. Robotics Research*, J.M. Brady and R. Paul, eds., MIT Press, Cambridge, MA, 413–424.

J.M. Brady, J. Ponce, A. Yuille and H. Asada (1985), "Describing surfaces", *Comput. Vision Graph. Image Processing*, **32**(1), 1–28.

R.A. Brooks (1981), "Symbolic reasoning among 3-D models and 2-D images", *Artif. Intell.*, **17**(1-3), 285–348.

B. Buchberger (1987), "Applications of Gröbner bases in non-linear computational geometry", *Trends in Computer Algebra: International Symposium*, R. Janssen, ed., Springer-Verlag, NY, 52–80.

P.J. Burt (1981), "Fast filter transforms for image processing", *Comput. Vision Graph. Images Processing*, **16**, 20–51.

J.F. Canny (1986), "A computational approach to edge detection", *IEEE Trans. Patt. Anal. Mach. Intell.*, **8**(6), 679–698.

J.F. Canny (1988), *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA.

M.P. do Carmo (1976), *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ.

D.A. Cyrluk, D. Kapur and J.L. Mundy (1988), "Algebraic reasoning in view consistency and parameterized model matching problems", *Proc. DARPA Image Understanding Workshop*, Boston, MA, 731–739.

A.L. Dixon (1908), "The eliminant of three quantics in two independent variables", *Proc. London Math. Soc. Series 2*, **7**, 49–69.

D. Eggert and K. Bowyer (1989), "Computing the orthographic projection aspect graph of solids of revolution", *Proc. IEEE Workshop on Interpretation of 3D Scenes*, Austin, TX, 102–108.

T.J. Fan, G. Médioni and R. Nevatia (1987), "Segmented descriptions of 3D surfaces", *IEEE Trans. Robotics and Automation*, **3**(6), 527–538.

T.J. Fan, G. Médioni and R. Nevatia (1988), "Matching 3D objects using surface descriptions", *Proc. IEEE Int. Conf. Robotics and Automation*, Philadelphia, PA, 1400–1406.

R.T. Farouki (1987), "Trimmed-surface algorithms for evaluation and interrogation of solid boundary representations", *IBM J. Research and Development*, **31**(3), 314–332.

O.D. Faugeras and M. Hebert (1986), "The representation, recognition, and locating of 3-D objects", *Int. J. Robotics Research*, **5**(3), 27–52.

R.S. Fearing (1987), *Tactile Sensing, Perception and Shape Interpretation*, PhD Thesis, Stanford University, Stanford, CA.

D. Forsyth, J. Mundy, A. Zisserman and C. Brown (1990), "Applications of invariant theory in computer vision", *Workshop on the Integration of Numerical and Symbolic Methods*, Saratoga Springs, NY. In this book.

R.N. Goldman and T.W. Sederberg (1985), "Some applications of resultants to problems in computational geometry", *The Visual Comput.*, **1**, 101–107.

W.E.L. Grimson (1990), *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA.

W.E.L. Grimson and T. Lozano-Pérez (1987), "Localizing overlapping parts by searching the interpretation tree", *IEEE Trans. Patt. Anal. Mach. Intell.*, **9**(4), 469–482.

A.D. Gross and T.E. Boult (1988), "Error of fit measures for recovering parametric solids", *Proc. Int. Conf. Comput. Vision*, Tampa, FL, 690–694.

K.T. Gunnarsson and F.B. Prinz (1987), "CAD model-based localization of parts in manufacturing", *IEEE Comput.*, **20**(8), 66–74.

M. Hebert and T. Kanade (1985), "The 3D profile method for object recognition", *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, San Francisco, CA, 458–463.

A. Hoogs (1991), *Computing the Pose of Curved Three-dimensional Objects from Range, Intensity or Contour Data*, Master Thesis, University of Illinois, Urbana-Champaign, IL.

R. Horaud (1987), "New methods for matching 3-D objects with single perspective views", *IEEE Trans. Patt. Anal. Mach. Intell.*, **9**(3), 401–412.

B.K.P. Horn (1984), "Extended Gaussian images", *Proc. IEEE*, **72**(12), 1671–1686.

D.P. Huttenlocher and S. Ullman (1987), "Object recognition using alignment", *Proc. Int. Conf. Comput. Vision*, London, UK, 102–111.

K. Ikeuchi and T. Kanade (1988), "Automatic generation of object recognition programs", *Proc. IEEE*, **76**(8), 1016–1035.

C. Jerian and R. Jain (1990), "Polynomial methods for structure from motion", *IEEE Trans. Patt. Anal. Mach. Intell.*, **12**(12), 1150–1166.

J.T. Kajiya (1982), "Ray tracing parametric patches", *Comput. Graph.*, **16**, 245–254.

J.J. Koenderink (1984), "What does the occluding contour tell us about solid shape?", *Perception*, **13**, 321–330.

J.J. Koenderink (1986), "An internal representation for solid shape based on the topological properties of the apparent contour", *Image Understanding: 1985-86*, W. Richards and S. Ullman, eds., Ablex Publishing Corp., Norwood, NJ, 257–285.

J.J. Koenderink and A.J. Van Doorn (1979), "The internal representation of solid shape with respect to vision", *Biological Cybernetics*, **32**, 211–216.

D.J. Kriegman (1989), *Object Classes and Image Contours in Model-Based Vision*, PhD Thesis, Stanford University, Stanford, CA.

D.J. Kriegman and J. Ponce (1990a), "Computing exact aspect graphs of curved objects: solids of revolution", *Int. J. Comput. Vision*, **5**(2), 119–135.

D.J. Kriegman and J. Ponce (1990b), "On recognizing and positioning curved 3D objects from image contours", *IEEE Trans. Patt. Anal. Mach. Intell.*, **12**(12), 1127–1137.

D.J. Kriegman and J. Ponce (1991), "Geometric modeling for computer vision", *SPIE Conf. Curves and Surfaces in Comput. Vision Graph. II*, Boston, MA, to appear.

D. Lowe (1987), "The viewpoint consistency constraint", *Int. J. Comput. Vision*, **1**(1), 57–72.

F.S. Macaulay (1916), *The Algebraic Theory of Modular Systems*, Cambridge University Press, Cambridge.

A.P. Morgan (1987), *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems*, Prentice Hall, Englewood Cliffs, NJ.

A.P. Pentland (1986), "Perceptual organization and the representation of natural form", *Artif. Intell.*, **28**, 293–331.

J. Ponce, D. Chelberg and W. Mann (1989), "Invariant properties of straight homogeneous generalized cylinders and their contours", *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(9), 951–966.

J. Ponce, A. Hoogs and D.J. Kriegman (1991), "On using CAD models to compute the pose of curved 3D objects", *IEEE Workshop on Directions in Automated "CAD-Based" Vision*, Maui, Hawaii, 136–145. Submitted to *Comput. Vision Graph. Image Processing.*

J. Ponce and D.J. Kriegman (1990a), "Computing exact aspect graphs of curved objects: parametric patches", *Proc. AAAI Conf. Artif. Intell.*, Boston, MA, 1074–1079.

J. Ponce and D.J. Kriegman (1990b), "New progress in prediction and interpretation of line-drawings of curved 3D objects", *Proc. 5th IEEE Int. Symp. Intelligent Control*, Philadelphia, PA.

W. Press, B. Flannery, S. Teukolsky and W. Vetterling (1986), *Numerical Recipes in C*, Cambridge University Press, Cambridge.

M. Raghavan and B. Roth (1989), "Kinematic analysis of the 6R manipulator of general geometry", *Proc. 6th Int. Symp. Robotics Research*, H. Miura and S. Arimoto, eds., MIT Press, Cambridge, MA, 314–320.

J.K. Salisbury (1984), "Interpretation of contact geometries from force measurements", *Proc. 1st Int. Symp. Robotics Research*, J.M. Brady and R. Paul, eds. MIT Press, Cambridge, MA, 565–577.

G. Salmon (1966), *Modern Higher Algebra*, Hodges, Smith, and Co., Dublin.

J. Schwartz and M. Sharir (1987), "On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds", *Planning, Geometry and Complexity of Robot Motion*, J. Schwartz, M. Sharir and J. Hopcroft, eds., Ablex Publishing Corp., Norwood, NJ, 51–96.

W.B. Seales and C.R. Dyer (1991), "Constrained viewpoint from occluding contour", *IEEE Workshop on Directions in Automated "CAD-Based" Vision*, Maui, Hawaii, 54–63.

T.W. Sederberg, D.C Anderson and R.N. Goldman (1984), "Implicit representation of parametric curves and surfaces", *Comput. Vision Graph. Images Processing*, **28**, 72–84.

T. Sripradisvarakul and R. Jain (1989), "Generating aspect graphs of curved objects", *Proc. IEEE Workshop on Interpretation of 3D Scenes*, Austin, TX, 109–115.

G. Taubin (1990), *Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations, with Applications to Edge and Range Image Segmentation*, Technical Report LEMS-66, Brown University, Providence, RI.

G. Taubin and D.B. Cooper (1990), "Recognition and positioning of 3D piecewise algebraic objects using Euclidean invariants", *Workshop on the Integration of Numerical and Symbolic Comput. Methods*, Saratoga Springs, NY. In this book.

D.W. Thompson and J.L. Mundy (1987), "Three-dimensional model matching from an unconstrained viewpoint", *IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 208–220.

R. Vaillant (1990), "Using occluding contours for 3D object modeling", *Proc. European Conf. Comput. Vision*. The journal version of this paper is R. Vaillant and O.D. Faugeras (1992), "Using external boundaries for 3D object modeling", *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(2), 157–173.

Appendix: Resultants

Bezout Resultants

We sketch Bezout's method for eliminating one variable among two equations (see also Sederberg *et al.*, 1984, for an elementary introduction). Consider two polynomials P_1, P_2 of degree n in x. We seek a necessary and sufficient condition for these two polynomials to admit a common root. Consider the determinant:

$$D(x,y) = \begin{vmatrix} P_1(x) & P_2(x) \\ P_1(y) & P_2(y) \end{vmatrix}. \tag{9.1}$$

This determinant vanishes whenever x is a common root of P_1, P_2 (the first row vanishes) and whenever $x = y$ (the two rows are identical). It follows that the polynomial D is divisible by $(x - y)$, and that the polynomial:

$$F(x,y) = D(x,y)/(x-y) \tag{9.2}$$

vanishes whenever x is a common root of P_1, P_2. Clearly, F has degree $n-1$ in x and y.

We can rewrite F as a polynomial in y, so that:

$$F(x,y) = \sum_{i=0}^{n-1} F_i(x) y^i, \tag{9.3}$$

where F_i is a polynomial of degree $n-1$ in x.

Now consider a common root $\hat{x}$ of P_1, P_2. For any value y, we have $F(\hat{x}, y) = 0$, and it follows that all $F_i(\hat{x})$ are equal to zero. Let F_i^j denote the coefficient of degree j of F_i, we obtain the following linear system:

$$\begin{pmatrix} F_0(\hat{x}) \\ \vdots \\ F_i(\hat{x}) \\ \vdots \\ F_{n-1}(\hat{x}) \end{pmatrix} = M \begin{pmatrix} \hat{x}^0 \\ \vdots \\ \hat{x}^j \\ \vdots \\ \hat{x}^{n-1} \end{pmatrix} = 0, \tag{9.4}$$

where:

$$M = \begin{pmatrix} F_0^0 & \dots & F_0^j & \dots F_0^{n-1} \\ \vdots & & \vdots & \vdots \\ F_i^0 & \dots & F_i^j & \dots F_i^{n-1} \\ \vdots & & \vdots & \vdots \\ F_{n-1}^0 & \dots F_{n-1}^j & \dots & F_{n-1}^{n-1} \end{pmatrix}. \tag{9.5}$$

Considering the successive powers $\hat{x}^j$ as so many independent variables, it follows that this homogeneous linear system admits a non-trivial solution if and only if its determinant vanishes, i.e. $|M| = 0$. This determinant is Bezout's resultant.

Dixon Resultants

We now sketch Dixon's method for eliminating two variables among three equations. Details on the method can be found in Dixon (1908) (see also Sederberg *et al.*, 1984, for an elementary introduction). The method is a generalization of Bezout's resultant.

Consider three polynomials P_1, P_2, P_3 in two variables x_1, x_2, with highest degree n in x_1 and m in x_2. We seek a necessary and sufficient condition for these three polynomials to admit a common root. Consider the determinant:

$$D(x_1, x_2, y_1, y_2) = \begin{vmatrix} P_1(x_1, x_2) & P_2(x_1, x_2) & P_3(x_1, x_2) \\ P_1(y_1, x_2) & P_2(y_1, x_2) & P_3(y_1, x_2) \\ P_1(y_1, y_2) & P_2(y_1, y_2) & P_3(y_1, y_2) \end{vmatrix}. \tag{9.6}$$

This determinant vanishes whenever (x_1, x_2) is a common root of P_1, P_2, P_3 (the first row vanishes), and also whenever $x_1 = y_1$ (the first two rows are identical) or $x_2 = y_2$ (the last two rows are identical). It follows that the polynomial D is divisible by $(x_1 - y_1)(x_2 - y_2)$, and that the polynomial:

$$F(x_1, x_2, y_1, y_2) = D(x_1, x_2, y_1, y_2)/((x_1 - x_2)(y_1 - y_2)) \tag{9.7}$$

vanishes whenever (x_1, x_2) is a common root of P_1, P_2, P_3. Clearly, F has degree $n-1$

in x_1, $2m-1$ in x_2, $2n-1$ in y_1, and $m-1$ in y_2. We can rewrite F as a polynomial in y_1 and y_2, so that:

$$F(x_1, x_2, y_1, y_2) = \sum_{\substack{i=0 \\ j=0}}^{\substack{2n-1 \\ m-1}} F_{i,j}(x_1, x_2) y_1^i y_2^j, \tag{9.8}$$

where $F_{i,j}$ is a polynomial in x_1 and x_2, with degree in x_1 (resp. x_2) less than or equal to $n-1$ (resp. $2m-1$).

Now consider a common root $(\hat{x}_1, \hat{x}_2)$ of P_1, P_2, P_3. For any value y_1, y_2, we have $F(\hat{x}_1, \hat{x}_2, y_1, y_2) = 0$, and it follows that all $F_{i,j}(\hat{x}_1, \hat{x}_2)$ are equal to zero. Let $F_{i,j}^{k,l}$ denote the coefficient of degree k in x_1 and l in x_2 of $F_{i,j}$, we obtain the following linear system:

$$\begin{pmatrix} F_{0,0}(\hat{x}_1, \hat{x}_2) \\ \vdots \\ F_{i,j}(\hat{x}_1, \hat{x}_2) \\ \vdots \\ F_{2n-1,m-1}(\hat{x}_1, \hat{x}_2) \end{pmatrix} = M \begin{pmatrix} \hat{x}_1^0 \hat{x}_2^0 \\ \vdots \\ \hat{x}_1^k \hat{x}_2^l \\ \vdots \\ \hat{x}_1^{n-1} \hat{x}_2^{2m-1} \end{pmatrix} = 0, \tag{9.9}$$

where:

$$M = \begin{pmatrix} F_{0,0}^{0,0} & \dots & F_{0,0}^{k,l} & \dots F_{0,0}^{n-1,2m-1} \\ \vdots & & \vdots & \vdots \\ F_{i,j}^{0,0} & \dots & F_{i,j}^{k,l} & \dots F_{i,j}^{n-1,2m-1} \\ \vdots & & \vdots & \vdots \\ F_{2n-1,m-1}^{0,0} \cdots & & F_{2n-1,m-1}^{k,l} \cdots & F_{2n-1,m-1}^{n-1,2m-1} \end{pmatrix}. \tag{9.10}$$

Considering the successive powers $\hat{x}_1^k \hat{x}_2^l$ as so many independent variables, it follows that this homogeneous linear system admits a non-trivial solution if and only if its determinant vanishes, i.e. $|M| = 0$. This determinant is Dixon's resultant. Explicit formulas for the Dixon resultant can be found in Dixon (1908) and Sederberg *et al.* (1984). More general methods, capable of dealing with arbitrary numbers of polynomials and variables, can be found in Macaulay (1916), Buchberger (1987) and Canny (1988).