Robust Object Recognition Based on Implicit Algebraic Curves And Surfaces *

Daniel Keren, Jayashree Subrahmonia, David B. Cooper
Laboratory for Engineering Man/Machine Systems
Division of Engineering, Brown University, Providence, RI 02912

Abstract

Implicit higher degree polynomials in x, y, z (or in x, y for curves in images) have considerable global and semiglobal representation power for objects in 3D space. (Spheres, cylinders, cones and planes are special cases of such polynomials restricted to second degree.) Hence, they have great potential for object recognition and position estimation. In this paper we deal with two problems pertinent to using these polynomials in real world robust systems: 1) Characterization and fitting algorithms for the subset of these algebraic curves and surfaces that is bounded and exists largely in the vicinity of the data; 2) A Mahalanobis distance for comparing the coefficients of two polynomials to determine whether the curves or surfaces that they represent are close over a specified region. These tools make practical the use of *geometric invariants* for determining whether one implicit polynomial curve or surface is a rotation, translation, or an affine transformation of another [2]. Though this technology handles objects with easily detectable features such as vertices, high curvature points, and straight lines, its great attraction is that it is ideally suited to smooth curves and smooth curved surfaces which do not have detectable features.

1 Introduction

Implicit polynomial curves and surfaces of degree higher than two have great modeling power for complicated shapes and can be made to fit data very well, but their coefficients may be sensitive to small changes in the data. This poses a problem since we would like to compare curves and surfaces based on their polynomial coefficients or functions of the coefficients that represent only shape, i.e., that are invariant to object rotation, translation and stretching in two directions – general linear transformations. In this paper we present new approaches and tools to these problems which should permit robust 2D curve and 3D surface object recognition and position estimation based on the *polynomial coefficients only.* In particular, we introduce the class of implicit polynomials that represent closed curves or surfaces, exhibit a low computation cost algorithm for fitting such that the curve or surface exsists only in the general region of the data, illustrate the wide range of shapes that can be represented and illustrate the improved stability of the coefficients. Then, for any polynomial whether it represents a closed or an open unbounded curve or surface, we present and discuss a simple expression for the aposteriori probability distribution of its coefficients given the data set that is to be represented by a polynomial. Polynomial coefficient sensitivity to small changes in the data occurs when a data set does not sufficiently constrain the coefficients. Our development determines the subset of coefficient space that is constrained by the data, and provides the appropriate metric for polynomial curve or surface recognition based on the polynomial coefficient vectors.

*This work was partially supported by NSF Grant #IRI-8715774 and NSF-DARPA Grant #IRI-8905436

2 Description of Closed Objects Using Polynomials

2.1 Finding the Fitting Polynomial

In general, for a polynomial $f(x, y)$ to describe a closed object O with boundary B the following should hold:

1) The set $\{(x, y) : f(x, y) = 0\}$ is equal to B.
2) $(x, y) \in O$ iff $f(x, y) < 0$.

We shall refer to the set $\{(x, y) : f(x, y) = 0\}$ as the *zero set.* Note that polynomials with an unbounded zero set can describe curve patches, but in Section 2 we are interested in describing closed bounded objects.

Since second degree polynomials can describe only circles and ellipses, we proceed to higher degrees. The standard notation for a polynomial of degree n will be adopted: $f(x, y) = \sum_{0 \le i+j \le n} a_{ij} x^i y^j$. The following simple lemma shows that the next class in the polynomial hierarchy is not suitable for describing closed objects.

Lemma 1 *The zero set of a third (or any odd) degree polynomial is unbounded.*

Proof: [3].

Next on the list are fourth degree polynomials. Their zero set can be bounded – e.g. $x^4 + y^4 - 1 = 0$ – or unbounded, e.g. $x^4 - y^4 = 0$. It is not surprising that the high powers of the polynomial determine if its zero set is bounded or not. Let us call the fourth degree powers, $a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4$, the *leading form* of $f(x, y)$, or $f_4(x, y)$, and the

0-8186-2855-3/92 $3.00 ©1992 IEEE

sum of the lower powers – cubics,quadrics,linear terms and the constant – the *lower terms* or $f_3(x, y)$. Let us also define a polynomial to be *stably bounded* if a small perturbation of its coefficients leaves its zero set bounded. For reasons of numerical robustness we are interested only in stably bounded polynomials.

Theorem 1 *The zero set of $f(x, y)$ is stably bounded iff there exists a symmetric positive definite matrix A such that $f_4(x, y) = (x^2\ xy\ y^2)A(x^2\ xy\ y^2)^T$.*

Proof: For the full proof, see [3]. The easy part of the proof proceeds as follows: suppose such a matrix A exists. Then $(x^2\ xy\ y^2)A(x^2\ xy\ y^2)^T \geq \lambda(x^4+x^2y^2+y^4)$, where λ is the smallest eigenvalue of A. It is clear that as (x, y) approaches infinity $f_4(x, y)$ dominates $f(x, y)$, and that f approaches infinity. So the zero set has to be bounded.

Summerizing, given an object O with boundary B we look for a fourth degree polynomial $f(x, y)$ such that:

1) $f_4(x, y)$ can be expressed as
$(x^2\ xy\ y^2)A(x^2\ xy\ y^2)^T$
with A symmetric positive definite.

2) The zero set of $f(x, y)$ approximates B.

A good approximation to the distance between a point (x_i, y_i) and the zero set of an implicit function f, suggested by [4] and extended in [6], is $\frac{f^2(x_i,y_i)}{\nabla^2 f(x_i,y_i)}$ (where $\nabla^2 f(x_i, y_i)$ stands for the square of the norm of the gradient). So the expression to be minimized is

$$\sum_{(x_i,y_i)\in B} \frac{f^2(x_i, y_i)}{\nabla^2 f(x_i, y_i)} \qquad (1)$$

Taubin [6], in an extensive work on implicit curve and surface fitting, solves the unconstrained fitting problem by approximating Equation 1 with the expression

$$\frac{\sum_{(x_i,y_i)\in B} f^2(x_i, y_i)}{\sum_{(x_i,y_i)\in B} \nabla^2 f(x_i, y_i)} \qquad (2)$$

and then minimizing Equation 2 by generalized eigen vector techniques, followed by an iterative scheme for improving the polynomial fit. Taubin's work results in excellent fits, but he does not constrain the zero set to be bounded; hence the outcome of his fitting algorithm is that the zero set *contains B* but often has additional unbounded parts (see Figures 1,2). A simple example is that of a square; Taubin's algorithm describes it as the union of four straight lines, with the corresponding $f(x, y)$ equal to the product of the four linear polynomials describing these lines. So the square is represented as the union of the infinite extension of its edges.

The question is how to incorporate into Taubin's algorithm the condition that the zero set be bounded. What should be done is simple: look only for polynomials $f(x, y)$ such that $f_4(x, y)$ can be expressed as in Theorem 1. The question is how to parametrize positive definite matrices. We use the fact that if a matrix A is symmetric positive semi definite, it has a symmetric *square root.* Hence it is enough to look at all $f(x, y)$'s where $f_4(x, y)$ can be written as $(x^2\ xy\ y^2)C^2(x^2\ xy\ y^2)^T$ where C is symmetric. Thus the strategy chosen was to minimize the error measure of Equation 1 while conforming to the above condition. This is done by minimizing not over the space of unconstrained polynomials, but only over the space of $f(x, y)$ such that $f_3(x, y)$ is unconstrained and $f_4(x, y)$ is as above. Technically, we look for the optimal C (6 parameters) and $f_3(x, y)$ (10 parameters). Note that in Equation 1, fourth powers of the elements of C appear. This does make the minimization problem non-linear, but that seems a reasonable price to pay if one wants to enforce boundness of the zero set. Sometimes, the zero set is bounded but contains spurious parts. This problem was solved by adding ϵI to C^2, where I is the identity matrix and ϵ a small positive constant. This also guarantees that A is positive definite.

Another problem affecting the running time of the fitting algorithm is that the expression in Equation 1 is expensive to calculate. Most non-linear minimization techniques require computation of the function and its derivatives many times. If many points are present, this means computing the sum of the function over its gradient squared in all these points, requiring enormous time. However, it is possible to overcome this problem using the following iterative algorithm:

1) minimize $\sum_{(x_i,y_i)\in B} f^2(x_i, y_i)$. This is quite fast, because the sum of the squares of the polynomial at the points can be written as FMF^T where F is the vector of the polynomial's coefficients and M is a scatter matrix of the points [6]. This is much faster than using Equation 1 directly. Call the optimal polynomial $F_1(x, y)$.

2) Assign to each point p_i a weight $w_i = \frac{1}{\nabla^2 F_1(p_i)}$.

3) Minimize $\sum w_i F^2(p_i)$. This is also quick – it is exactly the same Process as in **1)**, with M replaced by a weighted scatter matrix.

4)Go back to **2)** and update the weights using the minimizer of **3)** instead of $F_1(x, y)$.

5) Iterate untill the error of fit, measured by Equation 1, doesn't decrease substantially. (Note that we are using Equation 1, but only a small number of times – usually less than 5 iteration are needed).

This is almost similar to the algorithms suggested by Taubin and Sampson, but the minimization is carried out only on positive definite matrices and the normalization is different.

Some examples are presented. In Figure 1, an assortment of objects, each described by a fourth degree polynomial, are presented. In Figure 2a, a bounded description of a vase resembling shape is presented, and the unbounded one in 2b. In 2c, robustness of the fitting algorithm under noise is demonstrated for the same shape. In Figures 3,4 and 5, 3D objects are fitted with polynomials - a pear, an eggplant, and a complicated shape with holes (data was collected using an

IBM cartesian robot with tactile sensing). The fitting algorithm for 3D data is similar to that for 2D data and is described in [3].

3 Asymptotic Parameter Distributions, Mahalanobis Distances, And Bayesian Recognition

This section addresses the problem of variability in the polynomial coefficients with small changes in the data set by formulating it within a probabilistic framework. If the polynomial coefficients vary considerably, so will the invariants that are functions of these coefficients, thus giving unreliable recognition results. Thus, the first problem is to get an estimate of the uncertainity in the polynomial coefficients. The second problem is to design a metric based on the polynomial coefficients for comparing two polynomial zero sets *over the region where the data exists.*

The input data here is a sequence of range data points, $\mathbf{Z}^N = \{Z_1, Z_2, \ldots, Z_N\}$, with $Z_i = (x_i, y_i, z_i)^t$.

Let $\boldsymbol{\alpha}$ denote the vector of coefficients of the polynomial $f(x, y, z)$ that describes the given object. We assume that the range data points $Z_1, Z_2, \ldots, Z_N$ are statistically independent, with Z_i being a noisy Gaussian measurement of the object boundary in the direction perpendicular to the boundary at its closest point. [1, 2].

Thus, the joint probability of the data points is

$$p(\mathbf{Z}^N \mid \boldsymbol{\alpha}) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp[-\frac{1}{2\sigma^2} \sum_{i=1}^{N} \frac{f^2(Z_i)}{\| \nabla f(Z_i) \|^2}] \tag{3}$$

The maximum likelihood estimate $\hat{\boldsymbol{\alpha}}_N$ of $\boldsymbol{\alpha}$ given the data points is the value of $\boldsymbol{\alpha}$ that maximizes (3).

A very useful tool for solving the problems of object recognition and parameter estimation is an asymptotic approximation to the joint likelihood function, (3), which can be shown to have a Gaussian shape in $\boldsymbol{\alpha}$ [1], i.e.,

$$p(\mathbf{Z}^N \mid \boldsymbol{\alpha}) \approx [p(\mathbf{Z}^N \mid \hat{\boldsymbol{\alpha}}_N)] \exp\{-\frac{1}{2}(\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_N)^t \Psi_N (\boldsymbol{\alpha}-\hat{\boldsymbol{\alpha}}_N)\} \tag{4}$$

where Ψ_N is the uncertainity matrix of $\hat{\boldsymbol{\alpha}}_N$ having i, jth component, $-\frac{\partial^2}{\partial\alpha_i \partial\alpha_j} ln\, p(\mathbf{Z}^N \mid \boldsymbol{\alpha}) \mid_{\boldsymbol{\alpha}=\hat{\boldsymbol{\alpha}}_N}$.

The aposteriori distribution of $\boldsymbol{\alpha}$ given the data, i.e., $p(\boldsymbol{\alpha} \mid \mathbf{Z}^N)$, is propotional to $p(\mathbf{Z}^N \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha})$.

$$p(\boldsymbol{\alpha} \mid \mathbf{Z}^N) \approx constant \times$$
$$p(\mathbf{Z}^N \mid \hat{\boldsymbol{\alpha}}_N) \exp\left[-\tfrac{1}{2}(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}_N)^t \Psi_N (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}_N)\right] p(\boldsymbol{\alpha}) \tag{5}$$

where $p(\boldsymbol{\alpha})$ is a prior distribution for $\boldsymbol{\alpha}$.

This distribution addresses the first problem because it tells us about the uncertainity in the polynomial coefficients given the data points. If the uncertainity is large, it implies that the coefficients are not reliable. Then, instead of using the existing measurements to recognize the object, the system can collect more data in order to improve the parameter estimates [5].

3.1 Mahalanobis Distance as a Comparison Measure for Polynomial Zero Sets

The scenario that we consider here is one where the database consists of a set of L objects labeled $l = 1, 2, \ldots, L$, and modeled by polynomials of degree n. Let $\boldsymbol{\alpha}^l$ be the parameter vector for object l. The optimum recognition rule is: 'choose l for which $p(\mathbf{Z}^N \mid \boldsymbol{\alpha}^l)$ is maximum'.

Using the asymptotic approximation, (4), we see that since $p(\mathbf{Z}^N \mid \hat{\boldsymbol{\alpha}}_N)$ is independent of l, an approximately equivalent recognition is : choose l for which (6) is minimum

$$(\boldsymbol{\alpha}^l - \hat{\boldsymbol{\alpha}}_N)^t \Psi_N (\boldsymbol{\alpha}^l - \hat{\boldsymbol{\alpha}}_N) \tag{6}$$

The advantage in using (6) is that the data is involved just once (not L times) to compute $\hat{\boldsymbol{\alpha}}_N$ and Ψ_N. Note that (6) is a Mahalanobis distance measure. Using this distance measure is equivalent to checking how well the data set $Z_1, Z_2, \ldots, Z_N$ is fit by the polynomial having coefficient vector $\boldsymbol{\alpha}^l$.

An explanation for why the Mahalanobis distance is the appropriate metric for comparing polynomial zero sets is that even though a polynomial of degree n may be necessary to approximate the data well, the data may not constrain the coefficients of the fitted polynomial completely. Then, many different coefficient vectors may result in essentially equally good polynomial fits to the data. The matrix Ψ_N weights the various directions in $\boldsymbol{\alpha}$ space proportional to the reliability of the estimated coefficients given the data.

The scenario discussed here is the simplest. More complex scenarios, where each object is characterized by a distribution, are given in [5].

3.2 Experimental Results

Space limitation permits showing only a few data sets. See [5] for more. The data sets correspond to handwritten characters and are all well fit by third degree polynomials. Figures 6(a) through 6(d) show the data sets and the polynomial fits for the objects ('e','s','t' and 'r') in the database. Figure 7 corresponds to another instance of the handwritten character 'r' (that looks very much different from the one in the database.) The Mahalanobis distance (6) of the coefficient vector for the best polynomial fit to this data set to the coefficient vectors for the best polynomial fits to the letters 'e','s','t' and 'r' in the database are 15.84, 13.97, 13.91 and 1.0 respectively. For computing the Mahalanobis distance, we scale all the data sets so that they all lie within a rectangle of the same dimension. We see that the Mahalanobis distance is an effective substitute for (3). Of course, an 'r' like the one in the database would produce a much smaller distance.

4 Geometric Invariants

If the object is a rotated, translated, or an affine transformed version of an object in the database, then object recognition based on shape and not coordinate system parameters is accomplished through use of geometric invariants – functions of the coefficients of the best fitting polynomial to the data. Reliability can now be had by using the stabilized coefficients from section 2, or by using a Mahalanobis distance *in invariants space* obtained from (5). (See [5]).

References

[1] R.M. Bolle and D.B. Cooper. On optimally combining pieces of information, with applications to estimating 3D complex-object position from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):619–638, September 1986.

[2] D. Keren, J. Subrahmonia, G. Taubin, and D.B. Cooper. Bounded and Unbounded Implicit Polynomial Curves and Surfaces, Mahalanobis Distances, and Geometric Invariants, for Robust Object Recognition. In *Proceedings, DARPA Image Understanding Workshop*, January 1992.

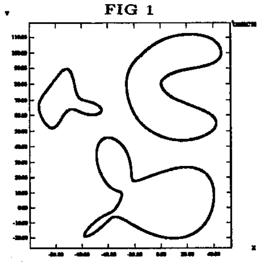
[3] D. Keren, D.B. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. Technical Report LEMS-93, Brown University, 1991.

[4] P.D Sampson. Fitting conic sections to very scattered data: an iterative improvement of the bookstein algorithm. *Computer Vision, Graphics, and Image Processing*, 18:97–108, 1982.

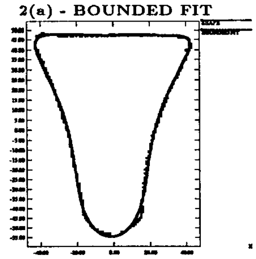
[5] J. Subrahmonia, D.B. Cooper, and D. Keren. Reliable object recognition using high dimensional implicit polynomials for 2D curves and 3D surfaces. Technical Report LEMS-94, Brown University, 1991.

[6] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1115—1138, November 1991.

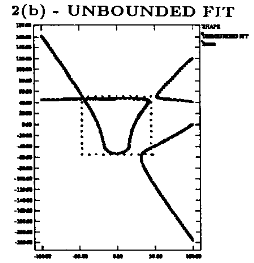
FIG 1



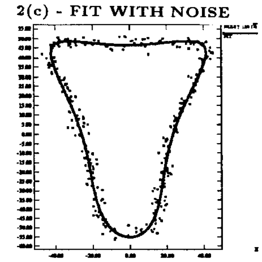
2(a) - BOUNDED FIT



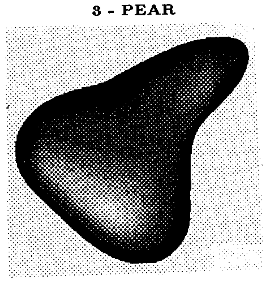
2(b) - UNBOUNDED FIT



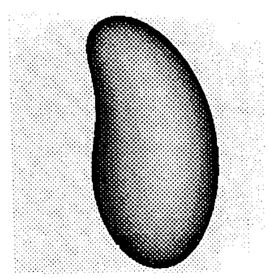
2(c) - FIT WITH NOISE



3 - PEAR



4 - EGGPLANT



5 - COMPLICATED SHAPE

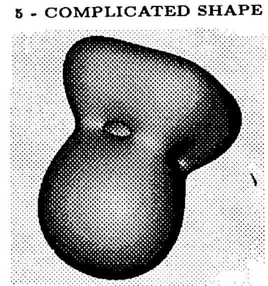


FIG 6(a) - 'e'

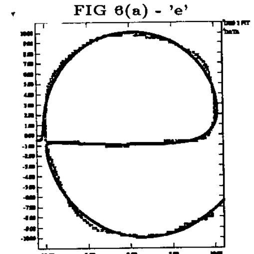


FIG 6(b) - 's'

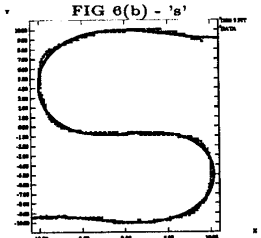


FIG 6(c) - 't'

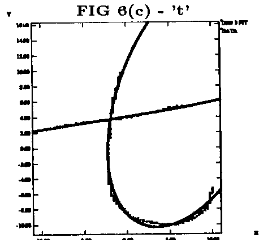


FIG 6(d) - 'r'

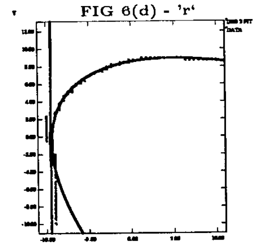


FIG 7