Describing Complicated Objects by Implicit Polynomials

Daniel Keren, David Cooper, *Senior Member, IEEE,* and Jayashree Subrahmonia

***Abstract*—This paper introduces and focuses on two problems. First is the representation power of closed implicit polynomials of modest degree for curves in 2-D images and surfaces in 3-D range data. Super quadrics are a small subset of object boundaries that are well fitted by these polynomials. The second problem is the stable computationally efficient fitting of noisy data by closed implicit polynomial curves and surfaces. The attractive features of these polynomials for Vision is discussed.**

I. Introduction and Previous Work

A. A Short Overview and Motivation

This work is to be combined with a general scheme for recognition and description of objects consisting of the following stages:

1) Segmentation into "volumetric primitives." This might be based on differential properties of the curve or surface [24], or on "interest regions" [23].
2) Fitting geometric models to each primitive. Super-quadrics have been widely in use for this [3], [24]. The model suggested here is closed implicit fourth-degree polynomials for 2-D curves and 3-D surfaces.
3) Recognition of objects using invariant features of the primitives, coupled with statistical techniques [11], [12], [21], [23]. For example, all or some of the primitives of a complex object could be represented by implicit polynomials, and the object indexed into a data base by aligning the primitives into a standard position or using polynomial invariants. This is treated in another paper.

This work concerns stage 2) above.

1) Why Fourth-Degree Polynomials?: This work is restricted to closed fourth-degree polynomials for the following reasons:

1) They are relatively simple and do not require an excessive number of coefficients to describe. There are 14 independent coefficients for 2-D curves and 34 for 3-D surfaces.
2) They are rich enough to contain other volumetric primitives that have been extensively used, such as conics and super-quadrics, and mathematically are easier to manipulate than super-quadrics. In this work we demonstrate that fourth-degree polynomials give good approximations to super-quadrics.
3) They can represent complex shapes including objects composed of a few disconnected components, objects that intersect themselves, and objects with holes.
4) Using richer geometric models, such as these polynomials, can make the difficult problem of segmentation and object recognition simpler, because an object does not have to be segmented into many patches each consisting of a simpler model.
5) The algorihtm used for fitting is very robust to noise.
6) Unlike splines, implicit polynomials can "fill in" where data is missing due to occlusion.

However, there is nothing unique about fourth-degree, and the algorithms presented here can be extended to any degree.

2) Main Contribution of this Work: Fitting implicit polynomials of degree higher than two is a relatively new area. The most complete work we know of is the excellent one by Taubin [20]. However, a difficulty with implicit polynomials is that their zero sets can be either unbounded or bounded but very large; in either case they describe the data well but their zero sets contain many points which are far away from the data points. In this work a paradigm for fitting polynomials whose zero set is *bounded, stable, and "tight"* around the object is presented. Also, an exact characterization of stably bounded curves is given. A nice range of shapes that can be represented by fourth-degree implicit polynomials having a bounded zero set is illustrated.

B. A General Overview

This paper introduces and focuses on two problems. The first is the representation power of implicit polynomials of modest degree for curves in 2-D images and surfaces in 3-D range data. Let $p(x, y)$ be a polynomial in x and y. By an implicit polynomial curve we mean the *zero set* $\{(x, y) : p(x, y) = 0\}$, and similarly for polynomials in x, y, and z. The second focus is the stable computationally efficient fitting of the most general *closed* implicit curves and surfaces to data that may be noisy.

The significance of the problems treated in this paper is the following. Implicit quadratic polynomials have proved to be very useful in computer vision for representing spheres, cylinders, cones, planes, and their 2-D corresponding curves

Manuscript received February 3, 1992; revised October 20, 1992. This work was supported by NSF Grant IRI-8715774 and NSF-DARPA Grant IRI-8905436. Recommended for acceptance by Associate Editor R. Nevatia.

D. Keren and D. Cooper are with the Laboratory for Engineering Man/Machine Systems, Division of Engineering, Brown University, Providence, RI 02912.

J. Subrahmonia was with the Laboratory for Engineering Man/Machine Systems, Division of Engineering, Brown University, Providence, RI 02912. She is now with IBM T. J. Watson Research Center, Hawthorne, NY.

IEEE Log Number 9212301.

in images [4], [6], [8]–[10], [15], [16], [18]. Now we go to a subset of the higher degree polynomials, namely the subset that represents closed curves and surfaces.

These implicit representations have their merits vis-a-vis explicit representations (e.g., splines) and the two complement each other nicely. Implicit representations are attractive for the following reasons:

1) The capability to describe irregulary shaped objects by a small number of parameters. For example, a popular tool for implicit representation in Vision are super quadrics [3], [7], [13]. In Sections II-B and III we show that super-quadrics are well represented by fourth-degree implicit polynomials, and, as can be seen from the examples in this paper, many shapes that can be represented by polynomials would be difficult to approximate even by the union of a few super-quadrics.
2) Object recognition by checking to see whether a data set is well fit by a specific implicit polynomial is fast. This is because there is a simple expression for very accurately approximating the distance from a data point to the zero set of the polynomial.
3) The coefficients of these polynomials when fit to data appear to be relatively insensitive to noise or to modest changes in the subset of the boundary used. Because of this stability, invariants of the coefficients (e.g., functions of the coefficients that are invariant to object translation, rotation, and general affine transformations) can be used for computationally fast object recognition. These Euclidean invariants are appropriate for recognizing objects in range data where object diameter is small compared to sensor distance, and affine invariants are appropriate for recognizing objects described by curves in aerial images when object diameter is small compared with object to camera distance and the objects are roughly planar, e.g., outline of an airplane on the ground. There are still occasions in which restricting the polynomial to be one having a bounded zero-set does not completely stabilize the coefficients. The Bayesian recognition and estimation techniques presented in [19] handle these residual cases easily.

   Other works have addressed the importance of invariants [11], [23].
4) The ability to conveniently translate operations in "object space" to natural operations in polynomials (for instance, union of objects corresponds to multiplication of polynomials).
5) Closed implicit polynomial fitting to noisy data has a low computational cost.
6) Implicit polynomials serve as *inside–outside functions* [3]: (x, y) is inside the object iff $p(x, y) < 0$. This provides a trivial rule for deciding whether a point is inside the object or not. Inside–outside functions are also very efficient in describing obstacles for robots and for many application in Graphics, e.g., ray tracing.
7) Closed "tight" polynomials faithfully describe the object, while unrestricted polynomials sometimes only contain the data.

The relation of the work in this paper to the existing technology is as follows.

There has been much more research on explicit representation, and only recently are implicit representations receiving the attention they deserve in Vision, Graphics, Robotics, and Computer Aided Geometric Design. In Vision, previous research was confined to fitting curves in the plane and surfaces in 3-D with conics, e.g., implicit polynomials of degree 2 [4], [6], [8], [10], [15], [16], [18] which are restricted. A breakthrough in using implicit polynomials in Vision is the work of Taubin [20], [23], where high degree polynomials are successfully fit to 2-D and 3-D data. Taubin also computes new invariants of polynomials.

Taubin's algorithm, however, suffers from the following problem: the zero set of the fitted polynomial is often unbounded. These unbounded fits *contain* the bounded data they are fit to, but take arbitrary shapes away from the data.

This poses two basic difficulties: first, the criterion for checking if a point belongs to an object loses much of its meaning, because the zero set and the object are not the same—the object is a small bounded portion of the zero set which is unbounded in the plane or in 3-D space. However, to test whether a point belongs to the object or not we substitute it in the polynomial, thus measuring its distance from the zero set. Second, we would like to recognize if two objects are the same by comparing the invariants of their polynomial representation. However, these are determined by the *global* behavior of the zero set, and we can run into situations where two polynomials that are totally different in the large will describe the same object—a highly undesirable phenomena.

We have taken two approaches to the recognition problem. In [19] we have shown that the stability problem results because not all terms in the polynomial are required for the representation. Hence, these can vary greatly with slight changes in the data set. The other approach, presented here, is to restrict consideration to the subset of implicit polynomials that have closed bounded zero sets. The added burden of being closed provides additional shape complexity that seems to constrain all the terms in the polynomial. Hence, our contribution here is in characterizing these closed implicit polynomials, and in introducing computationally fast fitting algorithms that provide stable fits.

It has recently come to our attention that Taubin *et al.* [22] have also been working on the topic of polynomials having closed zero sets. Although their work overlaps ours, their fitting algorithm is different and they do not treat the problem of necessary and sufficient conditions for a fourth-degree curve to have a stably bounded zero set. Also, they use a different minimization scheme, and a different method to minimize the area (or volume) of the resulting fit.

II. Description of Closed Objects Using Polynomials

A. Finding the Fitting Polynomial

A trivial but instructive example for defining objects by polynomials is the case of a circular object, say a circle of radius 1 with center at the origin (unit circle). For a point

(x, y) to be inside the circle it is necessary and sufficient that $x^2 + y^2 - 1 < 0$. Other objects that can be described by second-degree polynomials are ellipses. Hyperbolas, for instance, do not describe bounded objects—we need a closed bounded curve to do that! In general, for a polynomial $p(x, y)$ to describe an object O with boundary B the following should hold:

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

For a polynomial $p(x, y)$ we shall call the set $\{(x, y) : p(x, y) = 0\}$ the *zero set* of p. Thus the zero set of $x^2 + y^2 - 1$ is the unit circle.

Let us note at this point that polynomials with an unbounded zero set can describe curve patches, but in this work we are more interested in describing closed bounded objects. Bounded curves cannot *exactly* match some families of curves—like hyperbolas—but this is not crucial, because for every family of geometric primitives used, there are many objects that cannot be described *exactly*. The important issue is to have a family of primitives that is large and flexible.

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

Lemma 1: The zero set of a third-degree polynomial is unbounded.

Proof: Suppose that a_{30}, the coefficient of x^3, or a_{03}, the coefficient of y^3, is nonzero (if both are, a rotation of axis will make them nonzero). Suppose without loss of generality (WLG) that $a_{30} \neq 0$. Now take any y_0 and substitute it instead of y in the equation $p(x, y) = 0$. This results in a cubic equation in x, but every cubic equation has a real solution. This means that the zero set of $p(x, y)$ is unbounded in the y direction. It is trivial to verify that this simple proof carries on to any odd degree polynomial.

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 those powers, e.g., $a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4$, the *leading form* of $p(x, y)$, or $p_4(x, y)$, and the sum of the lower powers—e.g., cubics, quadrics, linear terms, and the constant—the *lower terms* or $p_3(x, y)$. 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 gives necessary and sufficient conditions for a fourth-degree polynomial to have a stably bounded zero set. The proof is given by way of a sequence of lemmas.

Lemma 2: If the leading form of $p(x, y)$ assumes both negative and positive values, the zero set of $p(x, y)$ is unbounded.

Proof: It is necessary to prove that outside of any circle around the origin there is a point in which $p(x, y)$ assumes a value of zero. Suppose there exist two points, (x_1, y_1) and (x_2, y_2) such that $p_4(x_1, y_1) > 0$ and $p_4(x_2, y_2) < 0$. Call the line going through the origin and (x_1, y_1) L_1, and the same for L_2. Noting that $p_4(\lambda x, \lambda y) = \lambda^4 p_4(x, y)$, it follows that moving far enough from the origin $p(x, y)$ will be positive on L_1 (this is because the leading form dominates the lower form as x and y grow larger) and negative on L_2. Take a point POS on L_1 that is outside of the previously mentioned circle such that p (POS)>0, and similarly NEG on L_2 with p(NEG) < 0. It is possible to connect POS and NEG with a curve that lies outside of the circle, and by the Mean-Value Theorem there is a point on that curve where $p(x, y)$ assumes zero. Since this can be carried out for every circle, the zero set is unbounded.

The next lemma shows that to insure stable boundness a stronger condition should be enforced:

Lemma 3: If the leading form $p_4(x, y)$ of $p(x, y)$ assumes a zero at a point other than the origin, then $p(x, y)$ is not stably bounded.

Proof: The proof is almost the same as for the previous lemma. Note that if $p_4(x, y) = 0$ then p_4 is zero on the line passing through the origin and (x, y). So, restricted to that line $p(x, y)$ is a cubic (in one variable) but the zero sets of cubics are never bounded. An exception can occur only if the cubic part of $p(x, y)$ is also zero on that line, but that is not a stable property; look, for example, at the polynomial $(x^2-y^2)^2+x^2+y^2-1 = x^4-2x^2y^2+y^4+x^2+y^2-1$. Its zero set is of course bounded—indeed it lies within the unit circle; but an arbitrarily small perturbation of this polynomial, e.g., $x^4 - 2x^2y^2 + y^4 + 10^{-10}x^3 + x^2 + y^2 - 1$, will make the zero set unbounded. Here the leading form does not assume both positive and negative values, so the previous lemma implies the zero set is bounded—alas not stably bounded, as $p_4(x, y)$ does assume a zero.

Formally, if the line in question is $y = bx$, the cubic part of the polynomial will assume a value of zero there iff $a_{30} + a_{21}b + a_{12}b^2 + a_{03}b^3 = 0$. Evidently this is not a stable property, as an infinitesimal change in the coefficients can cause b not to be a root of the cubic equation.

Lemma 4: If $p_4(x, y)$ is of constant sign and zero only at the origin, the zero set of $p(x, y)$ is stably bounded.

Proof: Assume WLG that the leading form is always positive (except at the origin where it is zero). Then it assumes some minimal (strictly positive) value on the unit circle (this is because the leading form is continuous and the unit circle is compact). Call this minimal value m. Since for every (x, y) the point $(x, y)/\|(x, y)\|$ is on the unit circle, it follows that $p_4((x, y)/\|(x, y)\|) \ge m$, therefore $p_4(x, y) \ge m(\|(x, y)\|)^4 = m(x^2 + y^2)^2$. This assures that $p(x, y)$ cannot have zeros whose distance from the origin is arbitrarily large, since again $p_4(x, y)$ dominates $p(x, y)$ for $x, y \to \infty$. Now, because of continuity, a small enough perturbation of the coefficients will leave $p_4(x, y)$ bounded away from zero on the unit circle, hence $p(x, y)$ is stably bounded.

Summarizing these simple proofs, we arrive at the conclusion that the zero set is stably bounded iff the leading form does not change sign. Let us assume WLG that if this is the case, it is always positive (note that multiplying the coefficients

by -1, or any constant for that matter, does not change the zero set).

In order to find conditions for stable boundness of the zero set, the following simple lemma is helpful:

Lemma 5: If the leading form of $p(x,y)$ can be written as

$$a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4 = (x^2xy\ y^2)A(x^2xy\ y^2)^T. \quad (1)$$

where A is a positive definite matrix, then the zero set is stably bounded.

Proof: As known from the theory of positive matrices [2], a positive definite matrix A satisfies $vAv^T \geq \lambda\|v\|^2$ for some positive constant λ. Using this together with (1) results in $p_4(x,y) \geq \lambda(x^4 + x^2y^2 + y^4)$. Therefore the leading form is positive except for the origin, and by Lemma 4 the zero set is stably bounded.

The converse is also true.

Lemma 6: If the zero set of $p(x,y)$ is stably bounded, there exists a positive definite matrix A such that $p_4(x,y) = (x^2xy\ y^2)A(x^2xy\ y^2)^T$.

Proof: As follows from Lemma 3, if the zero set is stably bounded, $p_4(x,y)$ does not have any real roots. Let us write

$$\begin{aligned} &a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4 \\ &= a_{40}y^4(X^4 + a_3X^3 + a_2X^2 + a_1X + a_0) \\ &= a_{40}y^4P(X), \end{aligned}$$

where $X = x/y$ and $a_3 = a_{31}/a_{40}$, etc. Since $P(X)$ does not have real roots, it can be written as found at the bottom of the page, where b and d are nonzero.

To prove that this 3×3 matrix is indeed positive definite, one can use the criterion of the diagonal submatrices [2], e.g., that the determinants of the 1×1, 2×2, and 3×3 submatrices along the diagonal are positive. But these determinants are equal to 1, $b^2 + d^2$, and $(abd - bcd)^2 + b^2d^4 + b^4d^2$, which are all positive (remembering that b and d are nonzero). To complete the proof, multiply by $a_{40}y^4$.

Note that the above matrix is also symmetric. All in all we have the following theorem

Theorem 1: The zero set of $p(x,y)$ is stably bounded iff there exists a symmetric positive definite matrix A such that $p_4(x,y) = (x^2xy\,y^2)A(x^2xy\,y^2)^T$.

An interesting fact is that this representation is nonunique. For instance,

$$x^4 - x^2y^2 + y^4 = (x^2xy\,y^2)\begin{pmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 \end{pmatrix}(x^2xy\,y^2)^T$$

and this matrix is not even positive semi-definite; however

$$x^4 - x^2y^2 + y^4 = (x^2xy\,y^2)\begin{pmatrix} 1 & 0 & -\frac{3}{4} \\ 0 & \frac{1}{2} & 0 \\ -\frac{3}{4} & 0 & 1 \end{pmatrix}(x^2xy\,y^2)^T$$

and that matrix is positive definite. However, the nonuniqness is of no concern at all, because it is the polynomial we are after, and the matrix A is not important as long as it gives the correct polynomial.

Summarizing, given an object O with boundary B, we look for a fourth-degree polynomial $p(x,y)$ such that

1) $p_4(x,y)$ can be expressed as $(x^2xy\,y^2)A(x^2xy\,y^2)^T$ with A symmetric positive definite;
2) the zero set of $p(x,y)$ approximates B.

We know what the first condition means. How is the second satisfied? The first guess is: find $p(x,y)$ such that $\sum_{(x_0,y_0)\in B} p^2(x_0,y_0)$ is minimal. This, however, results in a far from optimal description of B because $p^2(x_0,y_0)$ is a poor measure for the distance of (x_0,y_0) from the zero set of $p(x,y)$. A much better measure, suggested by [18] and extended in [23], is $p^2(x_0,y_0)/\nabla^2 p(x_0,y_0)$ (where $\nabla^2 p(x_0,y_0)$ stands for the norm of the gradient squared). So the expression to be minimized is

$$\sum_{(x_0,y_0)\in B} \frac{p^2(x_0,y_0)}{\nabla^2 p(x_0,y_0)}. \quad (2)$$

Taubin [23] solves the problem by approximating (2) with the expression

$$\frac{\sum_{(x_0,y_0)\in B} p^2(x_0,y_0)}{\sum_{(x_0,y_0)\in B} \nabla^2 p(x_0,y_0)} \quad (3)$$

and then minimizing (3) by generalized eigenvector techniques, followed by an iterative scheme based on (2) for improving the polynomial fit. Taubin's work results in excellent fits, but he does not worry about the zero set being bounded; hence a common outcome of his fitting algorithm is that the zero set *contains* B but often has additional unbounded parts (see Fig. 1). A simple example is that of a square; Taubin's algorithm describes it as the union of four straight lines, with the corresponding $p(x,y)$ 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. Naturally, this is not suitable for object description.

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 $p(x,y)$ such that $p_4(x,y)$ can be expressed as in Theorem 1. The question is

$$\begin{aligned} P(X) &= (X - a - bi)(X - a + bi)(X - c - di)(X - c + di) \\ &= (X^2X1)\begin{pmatrix} 1 & -(a+c) & ac \\ -(a+c) & (a+c)^2 + b^2 + d^2 & -(a^2c + b^2c + ac^2 + ad^2) \\ ac & -(a^2c + b^2c + ac^2 + ad^2) & (a^2+b^2)(c^2+d^2) \end{pmatrix}(X^2X1)^T \end{aligned}$$

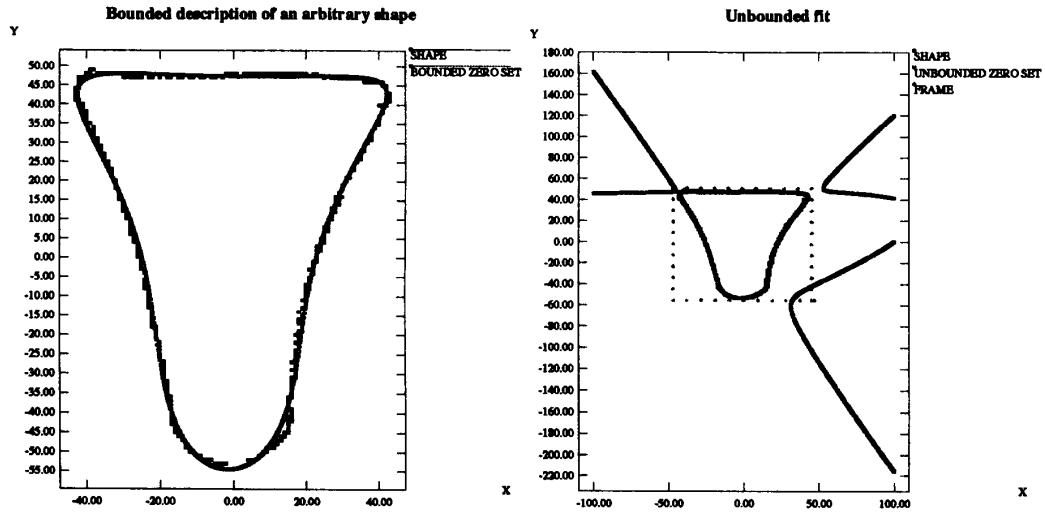


Fig. 1. Bounded versus unbounded fit.

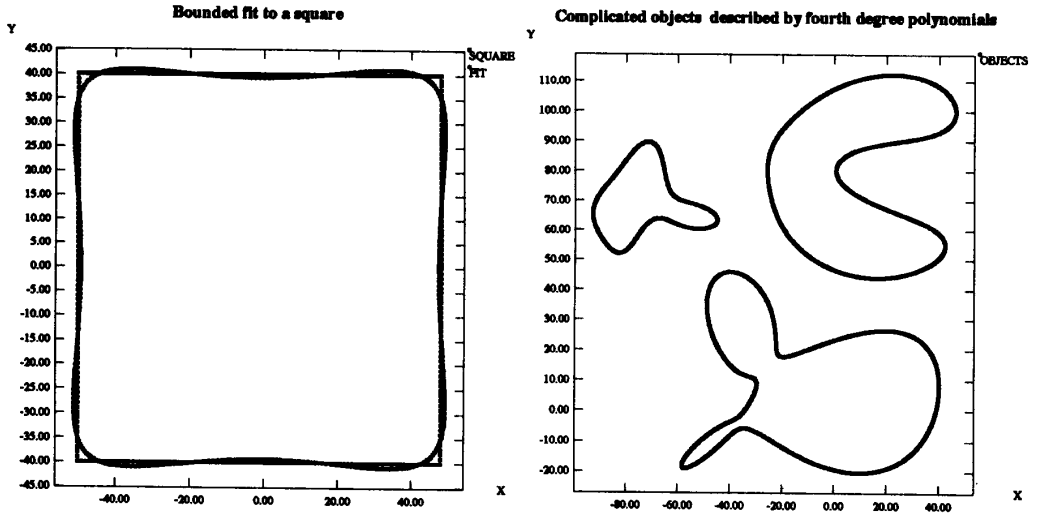


Fig. 2. Fits to square and arbitrary objects.

how to parametrize positive definite matrices. Taubin suggests using the following result [2]: if a matrix A is symmetric positive definite, it has a symmetric *square root* B.

Hence it is enough to look at all $p(x, y)$'s where $p_4(x, y)$ can be written as $(x^2 xy\, y^2)B^2(x^2 xy\, y^2)^T$ where B is symmetric. (Note that many B's can minimize the error function, but this does not matter.) Thus the strategy chosen was to minimize the error measure of (2) while conforming to the above condition. This is done by minimizing not over the space of unconstrained polynomials, but only over the space of $p(x, y)$ such that $p_3(x, y)$ is unconstrained and $p_4(x, y)$ is as above. Technically, we look for the optimal B (six parameters) and $p_3(x, y)$ (ten parameters). Note that in (2), fourth powers of the elements of B appear. This does make the minimization problem nonlinear, but that seems a reasonable price to pay in order to enforce boundness of the zero set. Following other work [6], [8], [23] we constrain the coefficients. The constraint was chosen to be independent of Euclidean transformations, and in the case of fourth degree curves amounts to

$$\frac{a_{40}^2}{24} + \frac{a_{31}^2}{6} + \frac{a_{22}^2}{4} + \frac{a_{13}^2}{6} + \frac{a_{04}^2}{24} = 1$$

because the above is a Euclidean invariant of the polynomial.

It is possible to reduce the number of parameters by one using the fact that if $p(x, y)$ is stably bounded then, following Lemma 6, $p_4(x, y)$ decomposes as

$$K[x - (a + bi)y][x - (a - bi)y][x - (c + di)y][x - (c - di)y]$$

for some K, a, b, c, d. This results in five parameters compared to the six of the symmetric matrix B. More tests have to be performed to compare these two ways of parametrizing stably bounded polynomials. In the Appendix we mention a third way

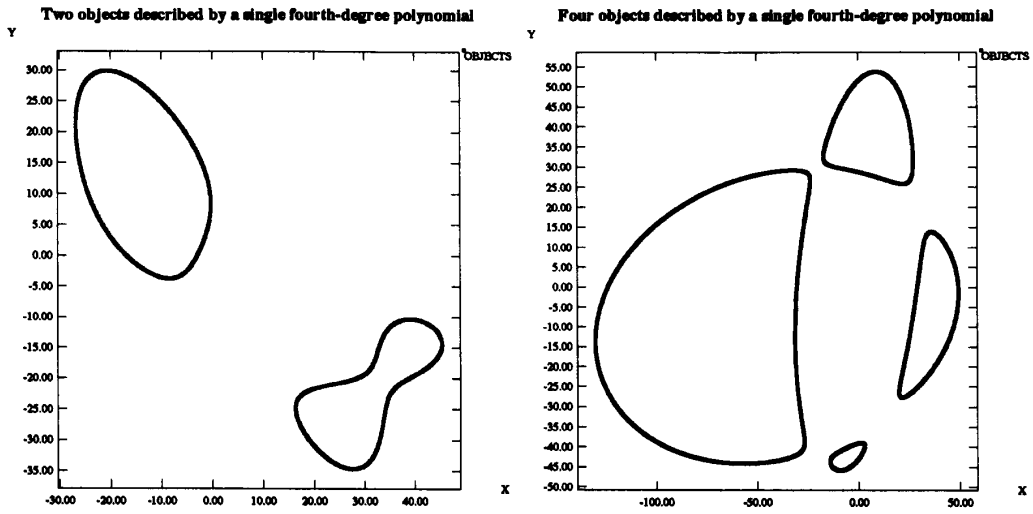


Fig. 3. Fitting a few objects with one polynomial.

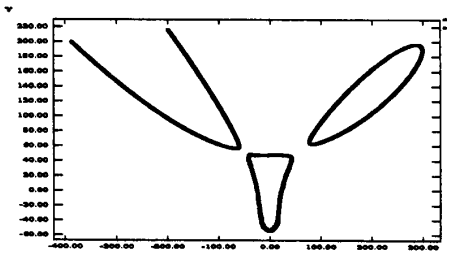


Fig. 4. Bad choice of epsilon.

of forcing $p_4(x, y)$ to be of the form $(x^2xy\,y^2)A(x^2xy\,y^2)^T$ for a positive definite A.

Another problem affecting the running time of the fitting algorithm is that the expression in (2) is expensive to calculate. Most nonlinear minimization techniques require numerous computations of the function and its derivatives. 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 $\sum_{(x_0,y_0)\in B} p^2(x_0, y_0)$. 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 [23]. This is much faster than using (2) directly. Note that this step incorporates the condition that the fit will be bounded, so it minimizes over the elements of a symmetric matrix that is raised to the second power. Since the distances are also squared, the expression to be minimized has degree four in the elements of the original matrix. Call the optimal polynomial $P_1(x, y)$.
2) Assign to each data point p_i a weight $w_i = \frac{1}{\nabla^2 P_1(p_i)}$.
3) Minimize $\sum w_i P^2(p_i)$. 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 $P_1(x, y)$.
5) Iterate until the error of fit, measured by (2), does not decrease substantially.

(Note that we are using (2), but only a small number of times—usually less than five iterations are needed).

This is almost identical to Taubin's and Sampson's algorithm; however, we use a different normalization of the coefficients, and force the resulting polynomial to be bounded.

This algorithm is suboptimal in the sense that it does not minimize (2), but it is much faster and results in the same quality of fits. Running times will be discussed in Section III.

In some cases the fitting algorithm presented here resulted in zero sets that are bounded but much larger than the fitted object. A simple heuristic to overcome this problem was to look at matrices not of the form B^2, but $B^2 + \epsilon I$, where ϵ is a small positive number and I the 3×3 identity matrix. This resulted in "tighter" fits, the reason being that for all (x, y)

$$(x^2xy\,y^2)(B^2 + \epsilon I)(x^2xy\,y^2)^T \geq (\lambda + \epsilon)(x^4 + x^2y^2 + y^4)$$

(where λ is the square root of the smallest eigenvalue of B^2), and as the leading form is magnified by the addition of the ϵ, the zero set tends to be more compact. A large ϵ forces the fit to "stick" to the data and prevents spurious pieces of the zero set from occurring, and causes the fit to be of a small area or volume, thus contributing to stability in the case of small data sets and noise; if not for the ϵ, we might run into cases where fitting a very small piece of data will result in a very large fit (like fitting a conic to a small part of a large ellipse). The importance of adding ϵ is demonstrated in Fig. 4 for the 2-D case; if $\epsilon = 0$ is used, we get not only the vase resembling shape but two spurious curves. A similar phenomenon is demonstrated in Fig. 17 for the 3-D case.

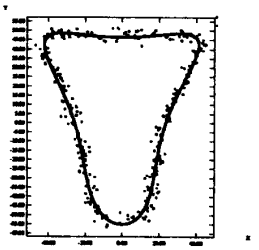


Fig. 5. Fit with noise.

The fitting procedure seems to be extremely robust to noise. In Fig. 5 the same data set as in Fig. 1 is presented with Gaussian noise of standard deviation 2 added, and with the appropriate fit.

Some examples are provided of bounded versus unbounded descriptions. In Fig. 1 an object is shown with a bounded and unbounded description. In Fig. 2 a bounded approximation of a square is presented. It is easy to see that no polynomial $p(x, y)$ with a bounded zero set can *exactly* describe the square (if it would, the polynomials defining the edges of the square would have to divide $p(x, y)$, hence the zero set would contain the infinite extensions of the square's edges). Also in Fig. 2, an assortment of objects that can be exactly described by fourth-degree polynomials is presented, and in Fig. 3 the power of polynomials is demonstrated by showing two and four disjoint objects that are represented by a single polynomial.

1) Choosing ϵ: The question arises how to choose a correct ϵ. If it is too small, it will not force the curve to be close to the data and we might end up with large spurious pieces of curve far away from the data. If it is too large, the zero set tends to shrink. Fortunately, all our experiments showed that there is enough freedom in the choice of ϵ so as not to make it a problem. If an object is centered around the origin, and its size is normalized so it lies in a 100×100 box, the choice of $\epsilon = 10^{-8}$ always worked (same for 3-D data).

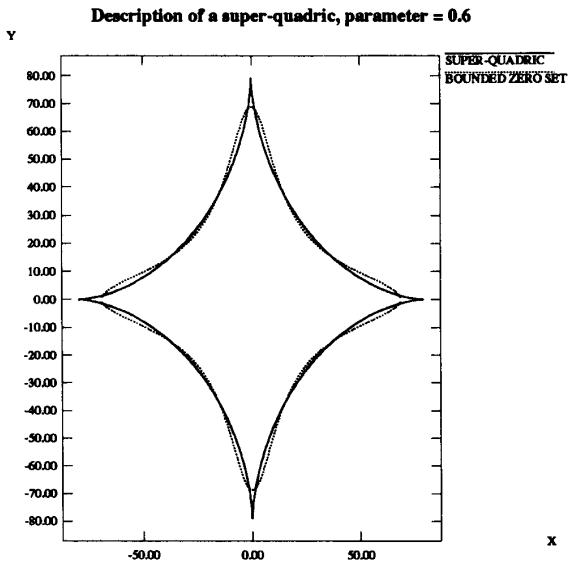
2) Description of 3-D Objects: Everything that was said about polynomials $p(x, y)$ extends to $p(x, y, z)$. Boundness of the zero set can be satisfied by using the identity, e.g., for a fourth-degree polynomial:

$$(x^2 xy\, xz\, y^2 yz\, z^2) A (x^2 xy\, xz\, y^2 yz\, z^2)^T = a_{400}x^4 + a_{310}x^3 y + a_{301}x^3 z + \ldots$$

where A is a 6×6 symmetric positive definite matrix, and proceeding with the same line of thought as for $p(x, y)$. In Section III results for 3-D fits will be surveyed.

B. Comparison to Other Models: Super-Quadrics

Another analytical model used to describe nonpolygonal objects in Graphics, Vision, and Robotics is the *super-quadric* [3], [7], [13]. As the name implies, these are extensions of ellipses and ellipsoids (and indeed are also called super-ellipsoids). In the plane, all super-quadrics can be described by a scaling, rotation, and translation of the "twisted circles" $C_\epsilon = \{(x, y) : x^\epsilon + y^\epsilon = 1\}$. If $\epsilon = 2$, C_ϵ is a circle; for



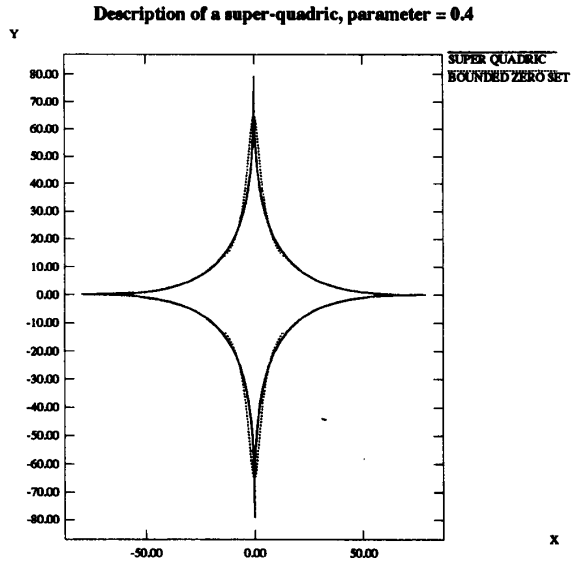


Fig. 6. Fits to super-quadrics.

$\epsilon > 2$, it is a circle that is "pushed out," approaching the unit square as ϵ grows; for small ϵ, C_ϵ is a nonconvex star-shaped set (see Fig. 6).

For description purposes, super-quadrics suffer from two limitations:

1) they are symmetric, while most objects are not;
2) they are too restricted, so usually a complicated object has to be broken into many disjoint parts, each of them described by a super-quadric.

They do have two advantages over implicit polynomials:

1) they have less degrees of freedom (6 for the 2-D case, 11 for 3-D);

2) they result in a representation that is both implicit and explicit.

Let us compare polynomials and super-quadrics. Since polynomials can also be scaled, rotated, and translated, then to compare the description power of polynomials to that of super-quadrics it is enough to check if the C_ϵ's are well approximated by polynomials. The approximations for $\epsilon = 0.6$ and $\epsilon = 0.4$ are presented in Fig. 6. For larger values of ϵ the approximations are so good that one cannot distinguish the super-quadric from the fitting polynomial by looking at them! These super-quadrics with larger ϵ, which are convex, are the ones used most frequently in Vision [5].

It can be seen that the polynomials give a reasonably good description of the super-quadrics, although they miss the sharp protrusions coming out of the sides of the super-quadrics; but that can be argued to be an asset, because we do not expect too many real objects to look like the super-quadric in Fig. 6. (Implicit polynomials can, however, represent spikes, as shown in Section III.)

Tests have thus led us to the conclusion that fourth-degree polynomials are a more general tool for describing objects than are super-quadrics. They are also simpler and do not involve transcendental functions as the super-quadrics do. Three-dimensional super-quadrics were also fit by polynomials; results are presented in Section III.

In general, it is impossible to describe a super-quadric *exactly* by an implicit polynomial. In [14] it is proved that if the shape parameter (e.g., ϵ) is rational, then the super-quadric can be represented exactly in this manner; however, the degree of the implicit polynomial might be very high. If one is willing to settle for a good approximation, fourth degree is enough.

C. Limitations of Fourth-Degree Polynomials

If polynomials of arbitrary degrees are allowed, then every object can be described, but it is better to work with degrees as low as possible. Let us state a famous theorem in algebraic geometry [1] that can help decide whether a fourth-degree polynomial can describe a given object.

Theorem 2: Beazout—If C_n and C_m are zero sets of polynomials of degree n and m that do not share a common component, they can intersect in at most nm points.

How is Beazout's Theorem useful? Suppose we are given an object O with boundary B, and observe that a line intersects B at five points. Because a line is described by polynomial of degree 1, the Theorem implies that O cannot be described by a fourth-degree polynomial. Fig. 7 gives an example that uses a circle (second-degree polynomial) to arrive at the same conclusion: the fact that there are ten intersection points means that to describe the object, a polynomial of degree at least 5 should be used.

In Fig. 3 it was demonstrated that a fourth-degree polynomial can describe four disconnected objects. This is an upper bound:

Lemma 7: A fourth degree polynomial cannot represent five (or more) disconnected closed components.

Proof: Suppose the contrary—that there is a fourth-degree implicit curve whose zero set is the union of boundaries

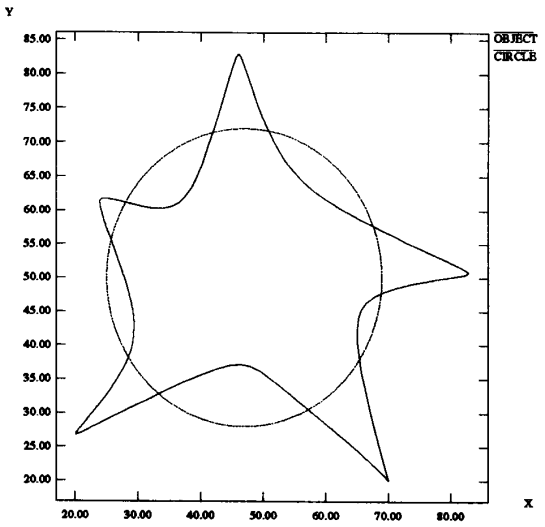


Fig. 7. An object that cannot be described by a fourth-degree polynomial.

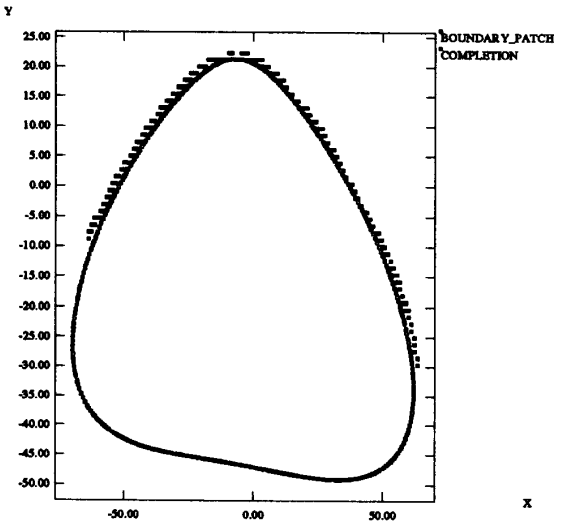


Fig. 8. Completion of a (magnified) boundary patch.

of five disconnected components. Then, choose a point inside each component. As is well known, there is a second-degree implicit curve that passes through these five points. Thus, the fourth-degree curve has to intersect the second-degree curve in ten points (going in and out of each component), a contradiction to Bezout's Theorem.

D. Behavior of Fitting Under Occlusion

A vision system usually does not have all the information on the objects in advance, and it gathers more and more information during its operation. A question thus arises how to relate to an object when only partial information about its boundary is given. This question does not have a deterministic answer. Suppose, for instance, that we are seeing some part of an object's boundary and it looks like half a circle. A natural

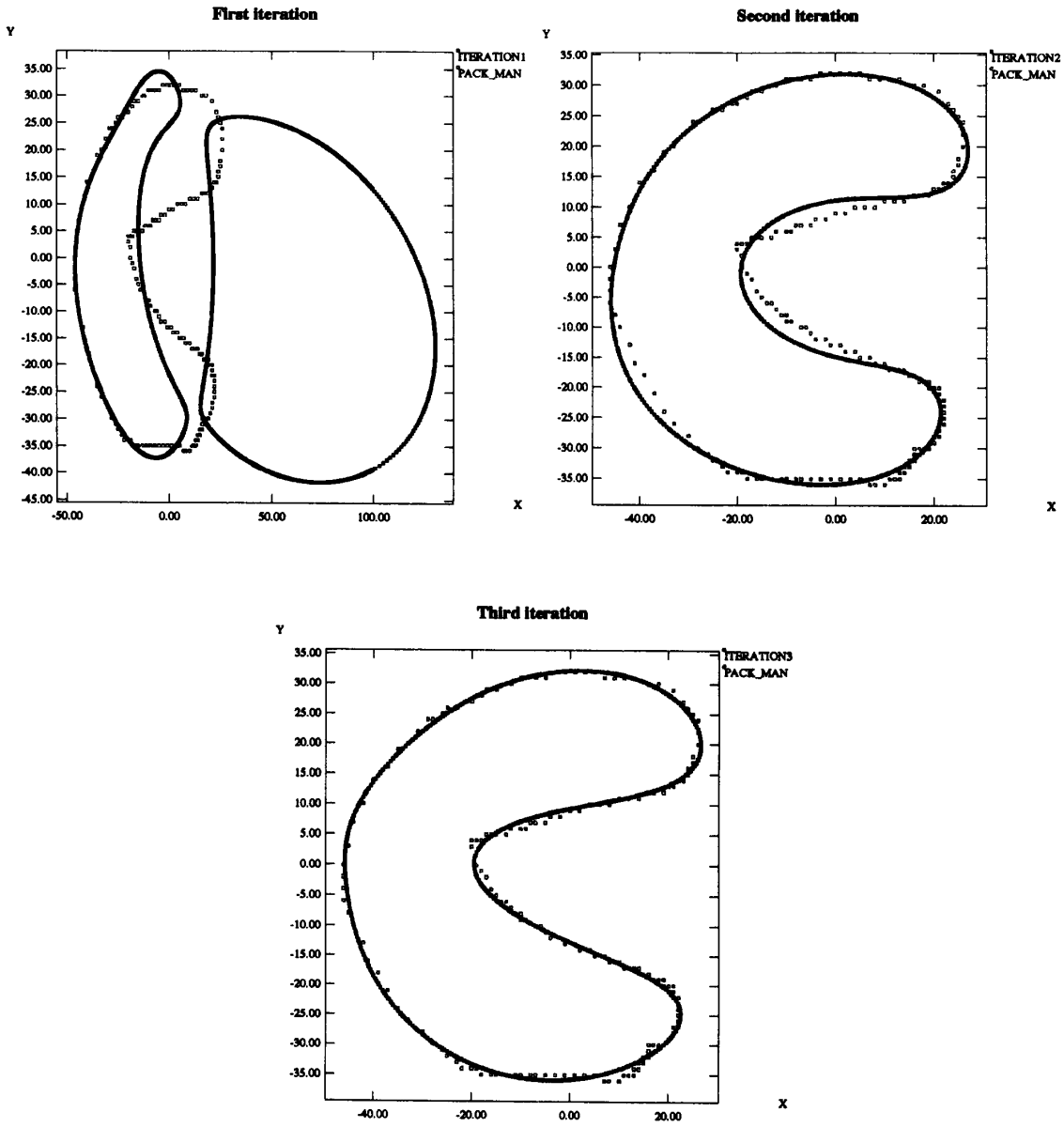


Fig. 9. Three iterations in the fitting procedure.

guess, consistent with the intuitive notion of symmetry, is that the object is a circle; but, of course, it may be many different objects of widely varying shape. What we should have is a scheme that "fills in" the missing part of an object while conforming to the following rules:

1) the object's boundary contains the observed boundary patch;
2) the object's boundary is smooth;
3) the object's area is not "too big."

Let us elaborate on condition 3. Suppose the partial information resembles a small patch of a very large circle. Symmetry would tell us that the object is this huge circle; but that may well lead to an object much larger than the real one.

Conditions 2 and 3 might loosely be described as instances of "Occam's Razor" rule: do not assume anything is complicated unless you have to.

We suggest that fitting the boundary patch with a bounded polynomial can reasonably fulfill conditions 1, 2, and 3. Experiments have led us to conclude that the great flexibility of polynomials allows them to "fill in" the missing part of the boundary while not generating an object of excessive area. Symmetric models, such as super-quadrics, might well satisfy conditions 1 and 2, but not 3.

TABLE I
FITTING TIME IN SECONDS

	Packman figure (175 points)		Two spheres (884 points)		Super–quadric I (441 points)		Super-quadric II (441 points)	
	time	error	time	error	time	error	time	error
Iteration 1	0.9	4.9	17.3	1.3	10.3	0.14	10.1	0.26
Iteration 2	1.1	1.4	43.2	0.000003	12.7	0.08	13.2	0.22
Iteration 3	1.4	0.68	-	-	-	-	15.7	0.15

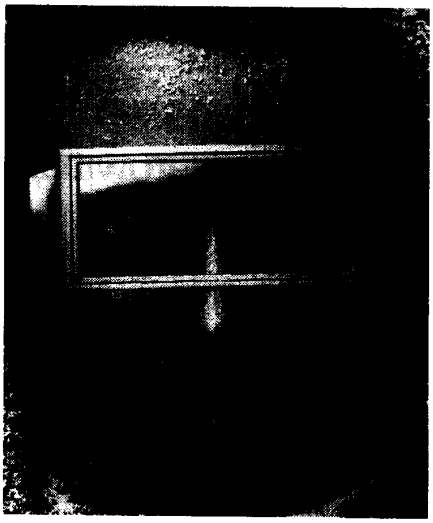


Fig. 10. Face.

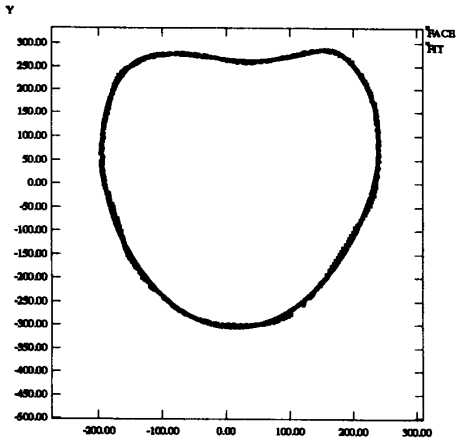


Fig. 11. Fit to face.

Hopefully, Fig. 8, which shows a boundary patch and its completion by a fourth-degree polynomial, will clarify this informal discussion. In that figure, a noisy incomplete boundary patch is shown. A symmetric or super-quadric model would probably "fill in" the missing part creating an area larger than the one resulting from description by a polynomial. The incomplete boundary piece is magnified by a factor of 1.1 in order to show the data points because in reality the polynomial description completely overlaps the boundary patch.

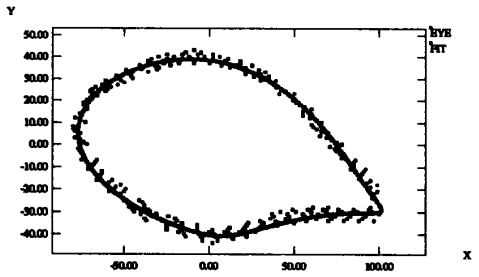


Fig. 12. Fit to eye.

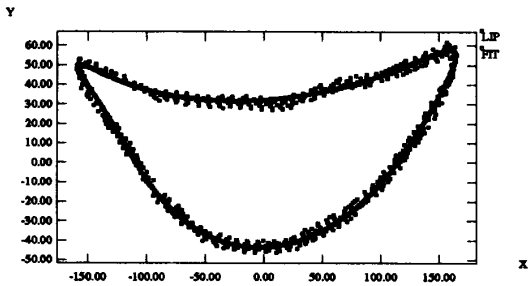


Fig. 13. Fit to lip.

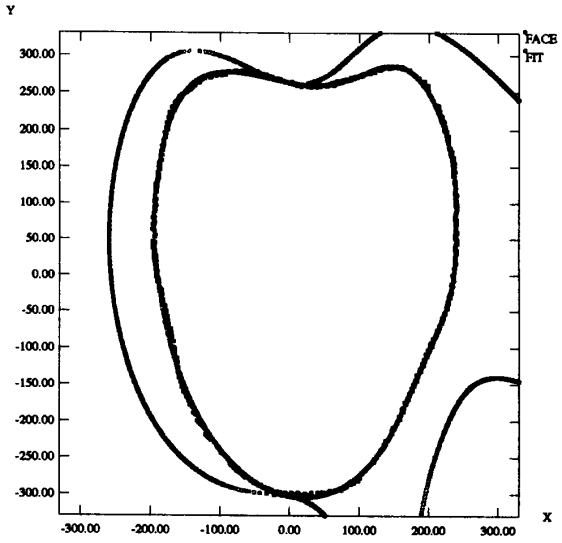


Fig. 14. Unrestricted fit to face.

Another point to observe is that as more and more information on the object's boundary is gathered, the describing polynomial can be computed quickly, because the polynomial

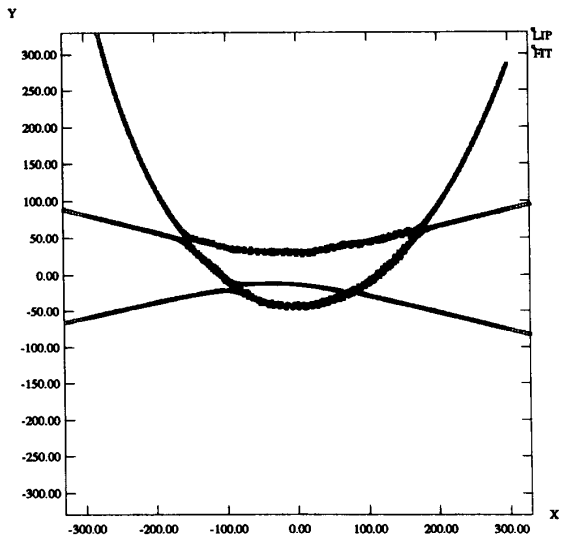


Fig. 15. Unrestricted fit to lip.

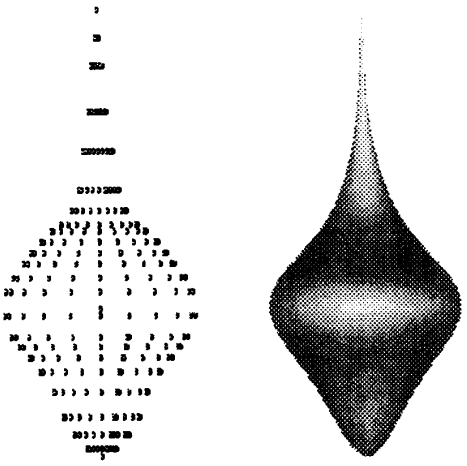


Fig. 16. Spike and fit to spike.

of the preceding boundary patch can be used as an initial approximation.

A similar attempt to obtain "volume minimizing fits" was carried out in [3] for super-quadrics, where the error function is multiplied by the volume of the fitted super-quadric, thus biasing the fit to have a smaller volume. In our case directly computing the volume is hard, but the addition of ϵ enables to control the volume without complicating the fitting procedure.

Another example of fitting to occluded data is given in the next section.

III. Experiments

Experiments were run on two- and three-dimensional data to test the fitting algorithm. The minimization scheme to solve the nonlinear optimization problem was Powell's method

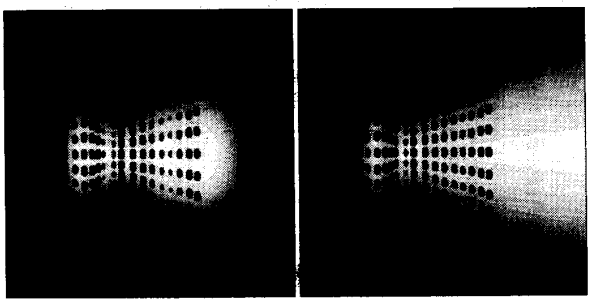


Fig. 17. Volume minimizing versus ordinary fit.

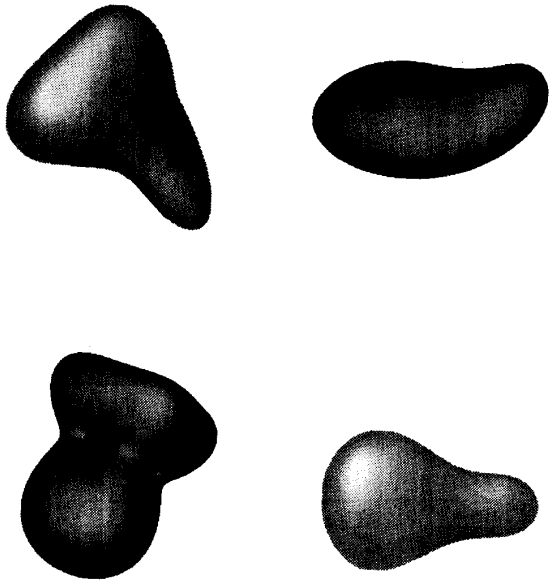


Fig. 18. Fit to 3-D data.

[17]. Implementation was quite simple, and the programs for fitting 2-D and 3-D data consist of about 500 and 800 lines in C, respectively, which were compiled and executed on a Sparc station. A nice feature of the algorithm is that no initial guess is needed—all iterations started with the zero polynomial.

Results are presented for one 2-D data set and three 3-D data sets. For each data set we present the accumulating running time (seconds) with respect to iterations, and the error of fit after that iteration. That error was defined to be the average distance of the data points from the fitted curve or surface.

The first example consists of 175 points in the shape of a packman. The first three iterations are shown in Fig. 9 superimposed on the data. After the third iteration, there is no significant improvement in the error. The error is larger than in the other examples, but that is because of the very nonsmooth nature of the data points.

The third example consists of 442 points that lie on a sphere of radius 50 around the origin and 442 points on a sphere of

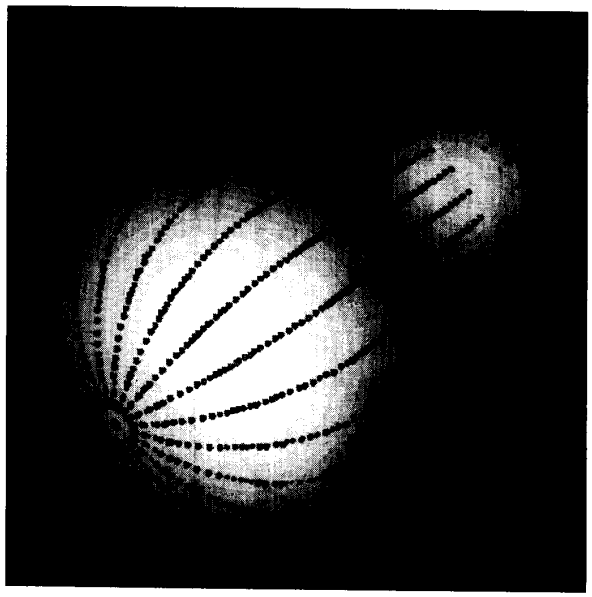


Fig. 19. Data points and fit to lightbulb.

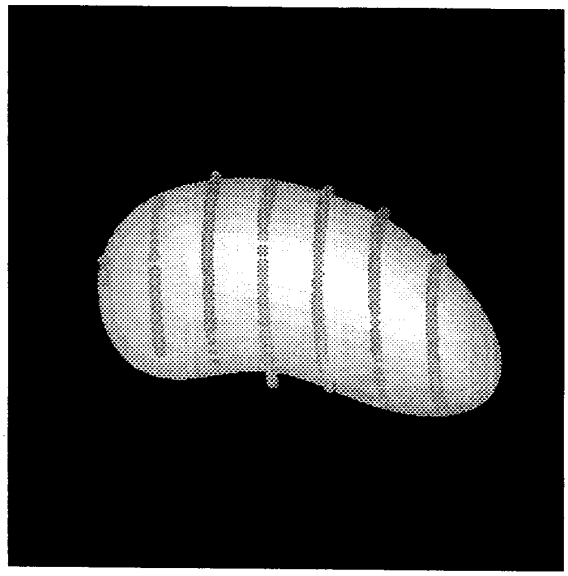


Fig. 20. Occluded data and fit for eggplant.

radius 100 around the origin. Since the union of the spheres can be exactly described by an implicit polynomial of degree four, the resulting error is very small.

The fourth example (super-quadric I) consists of 441 points on the surface of a super-quadric parametrized as

$$x = 50\cos^{0.4}(\phi)\cos^{0.6}(\theta)$$
$$y = 70\cos^{0.4}(\phi)\sin^{0.6}(\theta)$$
$$z = 100\sin^{0.4}(\phi).$$

The fourth example (super-quadric II) consists of 441 points

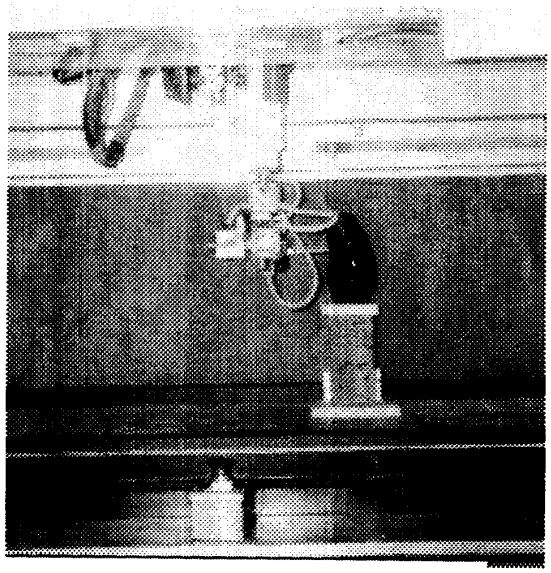


Fig. 21. Cartesian robot sensing data.

on the surface of a super quadric parametrized as

$$x = 50\cos^{1.8}(\phi)\cos^{2.8}(\theta)$$
$$y = 70\cos^{1.8}(\phi)\sin^{2.8}(\theta)$$
$$z = 100\sin^{1.8}(\phi).$$

It is known [3] that super-quadric I is a convex shape that looks like an ellipsoid "pushed out" and super-quadric II is a star-shaped object with spikes (compare Fig. 6 for the 2-D case) so it is not surprising that the approximation to super-quadric I is better (see Table I). Compared to the size of the super quadrics, the errors are reasonable. In [3] the time given to fit a super-quadric to 100 points is one minute on a VAX 785; the computer used in our experiments is faster. Considering that the space of fourth-degree polynomials is much larger than the space of super-quadrics, running times are reasonable.

The computation of the error function, and hence the fitting, can be easily made to run is parallel.

Some examples of real data are now presented. In Figs. 11–13 some parts of a human face are fitted by fourth-degree polynomials; in Fig. 10, the face is shown. In Fig. 11, the edge pixels of the face's contour are given super-imposed with the fit. In Figs. 12 and 13 the data for the eye and for the lower lip are given with the fits superimposed.

In order to show how crucial the bounded fitting is, we computed unrestricted fitting for the face and the lip, and the results are presented in Figs. 14 and 15. Once again, the fits contain the data but have spurious "branches."

Next, some 3-D examples are given. In Fig. 16 3-D sparse data are given together with their polynomial fit. Note that the "spike" is well fit by the polynomial, although quantization problems do affect the display. In Fig. 17, the importance of adding ϵ to the diagonal of B^2 (Section II-A) is demonstrated. When ϵ was not added, this resulted in a zero set that is much

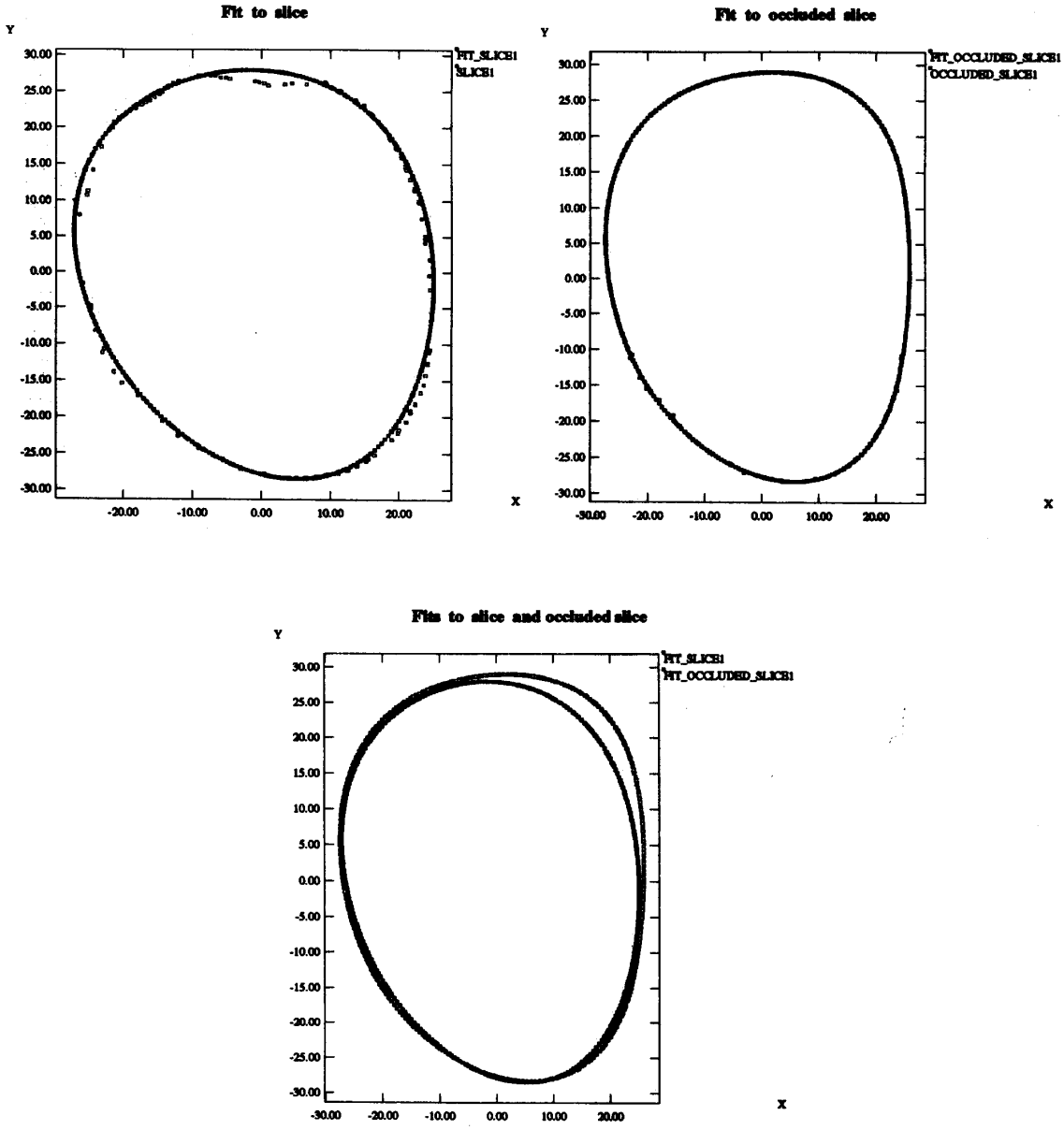


Fig. 22. Slices through fits to data and occluded data, I.

larger than the extent of the data. As noted, this is a common problem in fitting implicit models, such as super-quadrics. In Fig. 18 fits for data from the surface of a pear, an eggplant, and a light bulb are given, as well as as a complicated zero set with holes. In Fig. 19 the fit for the lightbulb is given with the data points superimposed on the surface.

In Fig. 20, it is demonstrated how polynomials "fill in" missing information. We tried to fit not the complete data for the eggplant, but only a part of it (shown superimposed as dots on the polynomial surface). The points shown are those having z values larger than 12 (the eggplant's points have z values ranging between -30 and 30). There are 288 points with these z values out of a total of 942 points on the eggplant's surface. In Figs. 22 and 23 we show how the occlusion affects the fits. Two slices through the eggplant data are shown, with slices through the fitted polynomial surface at corresponding heights. Together with these the slices of the occluded eggplant are presented; as can be seen there is much less data. The corresponding slices of the fit to the occluded eggplant are given. Finally, the fits to the complete data and the occluded data are given.

Data were collected in the LEMS laboratory using an IBM RS/1 Cartesian robot (Fig. 21). Running times were about the same as those given in Table I. The error of the fit was very

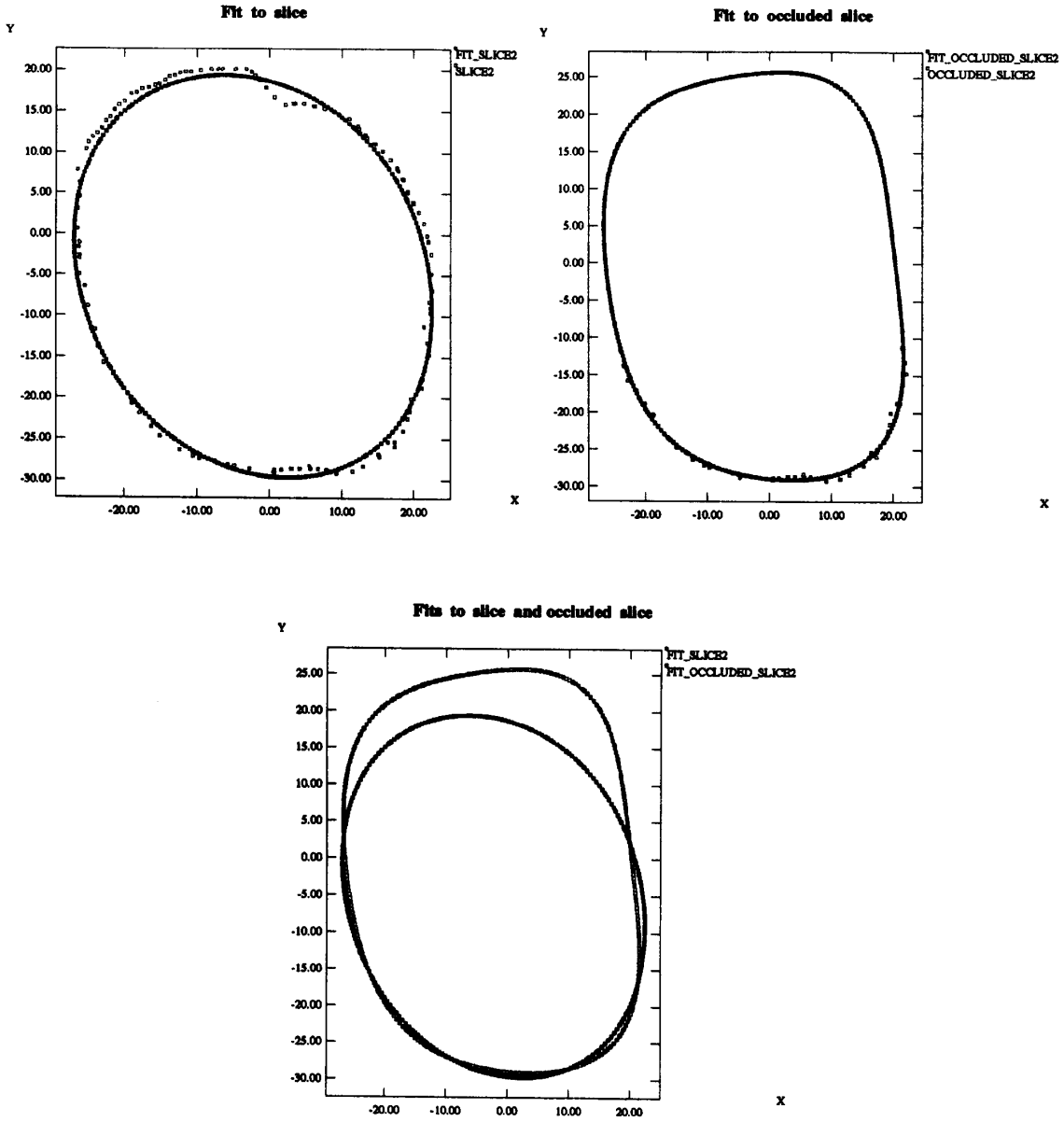


Fig. 23. Slice through fits to data and occluded data, II.

small: if the objects were normalized to lie in a $100 \times 100 \times 100$ cube, the typical error (distance of data from surface) was less than 0.3.

IV. Conclusion and Further Research

Of the three goals presented in the beginning of this paper (segmentation, description, and recognition), the second was described. It was demonstrated that unrestricted polynomial curves and surfaces can fit the data very well but usually contain points far removed from the given data set. The class of polynomial curves and surfaces introduced in this paper fit the data well and remain only in the vicinity of the data.

Many experiments were carried out on real 2-D and 3-D data illustrating the quality of the fits and the complexity of the shapes that can be represented. Further work is being conducted on the topics of segmentation and recognition of 2-D and 3-D data.

Appendix

In Section II, the question of how to force the leading term of the polynomial, $p_4(x, y)$, to be of the form $(x^2 xy\, y^2) A (x^2 xy\, y^2)^T$ for a positive definite A was addressed. In addition to the two methods suggested in Section II, a third that was tried is not to parametrize the space of

positive definite matrices, but to search for a fit in the space of all polynomials while adding to the error function of the fit a penalty term for matrices that are not positive definite. Such a penalty term is easy to construct, since a matrix is positive definite iff all the determinants on the diagonal are positive.

However, as we implemented it this method seemed to suffer from convergence problems, and many times got stuck in a matrix that was not positive definite (and resulted in an unbounded polynomial).

This can be attributed to the geometric structure of the space of coefficients.

Lemma 8: The space of coefficients of fourth-degree curves that are stably bounded, when viewed as a subspace of $\mathcal{R}^{15}$, is connected.

Proof: As noted in Section II, that space can be expressed as the continuous image of the space $\mathcal{R}^6 \times \mathcal{R}^{10}$, where $\mathcal{R}^6$ is identified naturally with the space of all symmetric matrices whose square gives the leading form, and $\mathcal{R}^{10}$ is identified with the lower degree terms. More explicitly, the function that assigns to any point in $\mathcal{R}^6 \times \mathcal{R}^{10}$ a polynomial is

$$\begin{aligned}&\{(p_1,p_2,p_3,p_4,p_5,p_6),(q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9,q_{10})\}\\&\quad\rightarrow a_1x^4+2a_2x^3y+(2a_3+a_4)x^2y^2+2a_5xy^3+a_6y^4\\&\qquad+q_1x^3+q_2x^2y+q_3xy^2+q_4y^3+q_5x^2+q_6xy\\&\qquad+q_7y^2+q_8x+q_9y+q_{10},\end{aligned}$$

where the matrix with elements $a_{i,j}$ is the square of the matrix with elements $p_{i,j}$. It is trivial to verify that the operation of squaring matrices is continuous in the coefficients; this is with $(p_1,p_2,p_3,p_4,p_5,p_6)$ identified with the symmetric matrix

$$A=\begin{pmatrix}p_1 & p_2 & p_3\\ p_2 & p_4 & p_5\\ p_3 & p_5 & p_6\end{pmatrix}$$

and using the representation $(x^2xyy^2)A(x^2xyy^2)$ for the leading form.

Now, this mapping is certainly continuous. Since the space $\mathcal{R}^6\times\mathcal{R}^{10}$ is connected, its continuous image is also connected.

However, although the space is connected, it is not linear, hence an optimization algorithm searching in that space can wander out of it and result in a point not belonging to it.

Acknowledgment

K. Korzeniowski proved invaluable help in using the Cartesian robot to collect real data. K. Ritter wrote the rendering program for implicit functions used to display the 3-D examples.

References

[1] S. S. Abhyankar. *Algebraic Geometry for Scientists and Engineers*. Providence, RI: American Mathematical Society, 1990.

[2] A. Albert. *Regression and The Moore-Penrose Pseudoinverse*. New York: Academic, 1972.

[3] R. Bajcsy and F. Solina, "Three-dimensional object representation revisited," in *Proc. Int. Conf. Computer Vision*, London, May 1987, pp. 231–240.

[4] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials," *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. PAMI-6, July 1984.

[5] R. M. Bolle and B. C. Vemuri, "Three dimensional surface reconstruction methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 1–14, Jan. 1991.

[6] F. L. Bookstein, "Fitting conic sections to scattered data," *Computer Vision, Graphics, and Image Processing*, vol. 9, pp. 56–71, 1979.

[7] T. E. Boult and A. D. Gross, "On the recovery of superellipsoids," in *Proc. DARPA Image Understanding Workshop*, 1988, pp. 1052–1063.

[8] B. Cernuschi-Frias, "Orientation and location parameter estimation of quadric surfaces in 3-D from a sequence of images," Ph.D. dissertation, Brown Univ., Providence, RI, 1984.

[9] D. B. Cooper and N. Yalabik, "On the computational cost of approximating and recognizing noise-perturbed straight lines and quadratic arcs in the plane," *IEEE Trans. Comput.*, vol. C-25, no. 10, pp. 1020–1032, Oct. 1976.

[10] O. D. Faugeras and M. Hebert, "Segmentation of range data into planar and quadric patches," in *Proc. CVPR*, 1983.

[11] D. Forsyth *et al.*, "Invariant descriptors for 3-D object recognition and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 971–992, Oct. 1991.

[12] D. A. Forsyth, J. L. Mundy, A. Zisserman, and C. M. Brown, "Projectively invariant representations using implicit algebraic curves," in *Proc. European Conf. Computer Vision*, June 1990.

[13] A. D. Gross and T. E. Boult, "Error of fit measures for recovering parametric solids," in *Proc. ICCV*, 1988, pp. 690–694.

[14] D. J. Kriegman and J. Ponce, "On recognizing and positioning curved 3-D objects from image contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 1127–1138 Dec. 1990.

[15] O. D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of range data into planar and quadratic patches," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Washington, DC, June 1983, pp. 8–13.

[16] V. Pratt, "Direct least squares fitting of algebraic surfaces," *Computer Graphics*, vol. 21, pp. 145–152, July 1987.

[17] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Receipes*. Cambridge, England: Cambridge University Press, 1986.

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

[19] J. Subrahmonia, D. Cooper, and D. Keren, "Reliable object recognition using high dimensional implicit polynomials for 2-D curves and 3-D surfaces," Brown Univ., Providence, RI, Tech. Rep. LEMS-94, 1991.

[20] G. Taubin, "Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation," Brown Univ., Providence, RI, Tech. Rep. LEMS-66, Jan. 1990; also in *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. 13, pp. 1115–1138, Nov. 1991.

[21] G. Taubin and D. B. Cooper, "2-D and 3-D object recognition and positioning system based on moment invariants," in *Proc. DARPA-ESPIRIT Workshop Geometric Invariants*, Rikjavik, Iceland, May 1991.

[22] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D. J. Kriegman, "Parameterizing and fitting bounded algebraic curves and surfaces," in *Proc. CVPR*, 1992, pp. 103–109.

[23] G. Taubin, "Recognition and positioning of rigid objects using algebraic and moment invariants," Ph.D. dissertation, Brown Univ., Providence, RI, 1990.

[24] P. Whaite and F. P. Ferrie, "From uncertainty to visual exploration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 1038–1049, 1991.