International Journal of Shape Modeling, Vol. 4, Nos. 3 & 4 (1998) 111–126

TIGHT FITTING OF CONVEX POLYHEDRAL SHAPES

CRAIG GOTSMAN
Faculty of Computer Science, Technion
Haifa, 32000 Israel

and

DANIEL KEREN
Department of Computer Science, University of Haifa
Haifa, 31905 Israel

Received 10 September 1997
Revised 11 May 1998

A problem which often arises while fitting implicit polynomials to 2D and 3D data sets is the following: although the data set is simple, the fit exhibits undesired phenomena, such as loops, holes, extraneous components etc. In addition to solving this problem, it is often desired to have a "tight fit" for a data set, i.e., a polynomial with a zero set which contains the data, but as little extra area (or volume) as possible. Such "tight fits" are of special interest in robotics (for compactly describing obstacles), and in computer graphics (for ray tracing and collision detection). Previous work tackled these problems by optimizing heuristic cost functions, which penalize some of these topological problems in the fit. This paper suggests a different approach – to design parameterized families of polynomials whose zero sets are *guaranteed* to satisfy certain topological properties. Namely, we construct families of polynomials with zero sets which are guaranteed to contain a given convex polyhedral shape, and which are also "tight" around it. The ability to parameterize these families depends heavily on the ability to parameterize positive polynomials. To achieve this, we use some powerful recent results from real algebraic geometry.

Keywords: Implicit Polynomials, Convex Polyhedra, Topological Integrity, Positive Polynomials.

1. Introduction

Fitting analytic functions to sampled data is a common problem arising in many data modeling applications. In its most general form, the fitting problem is: Given a set of n data points $S = \{\bar{x}^i = (x_1^i, .., x_d^i) \in \mathcal{R}^d : i = 1, .., n\}$, find an analytic surface that passes "close" to S. Common representations of such a surface are parametric surfaces defined on $\mathcal{R}^{d-1}$ or zero sets of a function $F : \mathcal{R}^d \rightarrow \mathcal{R}$. The latter is the locus of all points $\bar{x}$ such that $F(\bar{x}) = 0$. Candidates for F are any interpolant over $\mathcal{R}^d$, e.g. radial basis functions, super-quadrics, thin-plate splines, or polynomials. In the latter, the number of degrees of freedom, i.e. polynomial coefficients, is $m = \binom{d+k}{k}$, where k is the degree of the polynomial.

The advantages of using an implicit polynomial are its simplicity, the possibility to compute algebraic invariants [6, 5, 11, 8, 17] and the ease of containment computations (by computing the sign of the polynomial). Also, it is often desirable that a function which describes a given shape will be differentiable; this is very helpful, for instance, in obstacle avoidance algorithms which use variational principles and therefore have to compute the derivative of the obstacle's potential function [12].

The simplest way of fitting an implicit polynomial to the data set S is to solve the following least-squares problem for the coefficient vector $\bar{a}$ of the polynomial $P_{\bar{a}}$:

$$\bar{a} = arg\ min_{\bar{a} \in \mathcal{R}^m} \sum_{i=1}^{n} P_{\bar{a}}(\bar{x}^i)^2 \tag{1}$$

This cost function minimized in (1) is simple, therefore the problem may be solved easily by an eigenvector computation. However, the cost function does not necessarily express the Euclidean distances of the data points to the zero-surface, therefore the fit might be somewhat unintuitive, especially in regions of high surface curvature, as has been shown in previous works [20, 9, 19]. A cost function that approximates the sum of the squares of the Euclidean distances is:

$$\bar{a} = arg\ min_{\bar{a} \in \mathcal{R}^m} \sum_{i} \left(\frac{P_{\bar{a}}(\bar{x}^i)}{||\nabla P_{\bar{a}}(\bar{x}^i)||} \right)^2 \tag{2}$$

where $\nabla P(\bar{x}) = (\partial P/\partial x_1, .., \partial P/\partial x_d)$ is the vector gradient function. Unfortunately, this cost function induces a non-linear least-squares problem, whose numerical solution suffers from the usual non-linear optimization algorithmic pitfalls, namely, slow iterative solution, and local minima. If computation time is not a factor, as is the case in some applications, a solution to (2) is usually superior to that of (1). Because of its rational form, (2) may be solved iteratively as a sequence of *weighted* linear least squares problems [9, 19, 16, 3], which is a numerical procedure more efficient than general purpose optimization.

The disadvantages of using implicit polynomials as a modeling tool are the quality of the results commonly obtained when applying the above procedures, especially for high degrees. The zero sets may consist of multiple components, be unbounded, or fit the data in very unnatural ways [9, 20, 19]. It is very difficult to predict the outcome of the fitting procedure, a problem compounded by the fact that the polynomial coefficients are geometrically meaningless. A typical data-fitting session consists of running the optimization procedure (2) again and again, obtaining local minima, and choosing that yielding the best solution. Many trials may be required until a pleasing fit is found. In some cases, the procedure seems to never end, in the sense that the pleasing local minima are so sparse, that it is virtually impossible to stumble on them at random. An open question is how to restrict the search space to a subset of "well-behaved" polynomials, thus reducing the number of trials until success.

The main question is how to compactly represent, i.e. parameterize, this family of well behaved polynomials by imposing some restrictions on the polynomial coefficients. In [9, 20], it is shown how to guarantee that the zero set is bounded. The effort to achieve pleasing fits was carried further in [18], where it was suggested to use the geometric distance (as opposed to algebraic distance) in order to fit implicit polynomials. This resulted in better fits without holes. Also, a method to eliminate extraneous components was suggested. In [15], polynomials with a convex zero set are fitted to convex polygons, so that their zero set contains the polygon; the degree of the polynomial is equal to the number of vertices in the polygon. In [2], polynomials whose zero set is guaranteed not to have folds within a certain region are constructed, and many such "A-patches" are used to describe shapes. Here, "nice" behavior of the zero set is globally enforced. In work reported recently in [14], an attempt is made to force the zero-set to "stick" to the data, thus hopefully minimizing the number of branches etc. in the zero-set.

However, the algorithms presented in [9, 20, 18] are heuristic in nature. They try to force the resulting fits to have certain "good" geometric properties (such as being "tight" around the data set), by minimizing a cost function that penalizes fits which are "not good". In this work, we suggest a different approach – find an analytic parameterization of a (large as possible) sub-family of polynomials whose zero sets *always* have these "good" properties, and restrict the search for a pleasing fit to this sub-family. This guarantees that the fit will be "good", and eliminates the necessity of using a penalizing function.

A task of special importance is to force the zero set to contain the object, but to also be as "tight" around it as possible. This is important in various applications, such as the obstacle avoidance problem in robotics. If the description of the obstacles is not tight, this may result in the system choosing an inefficient path – as it assumes that the obstacles occupy a larger area/volume than they really do. The system may even fail to find a path in that case, even if one does exist. One must also guarantee that the bounding volume contain the obstacle, else the robot may collide with it.

In the realm of computer graphics, ray tracing algorithms usually try to filter out rays which do not intersect some bounding volume around the object. This can be done by using bounding ellipsoids, which are really a special case of bounding implicit polynomials, treated in this paper (quadratics) [1]. However, it should be noted that the volume of the best bounding ellipsoid grows at a super-exponential rate in the dimension of the space, relative to the convex set it bounds, if we take the supremum of the volume ratio over all convex sets. Therefore, in high dimensions bounding ellipsoids may not be appropriate to use as bounding volumes. Let us note that even if the underlying problem is two or three dimensional, it is often solved in a space of higher dimension (for instance, the configuration space of a rigid robot in three dimensional space is six dimensional, and it can be higher for more complicated robots). Therefore, it is important to find bounding volumes which are tighter than the quadratics.

In this paper we present a family of polynomials which give very tight bounds to convex polyhedra, while guaranteed to contain them.

2. The General Method

We are interested in restricting our search to a subset of the polynomials with given characteristics. The question becomes how to easily parameterize that subset. In general we will not be able to parameterize precisely the subset we are interested in, but only a smaller subset of it, since we are able to formulate only sufficient (but not necessary) conditions for a polynomial to have the required characteristics. These conditions lead to an unconstrained search on a parameter space, whose dimension might be larger than the dimension of the original polynomial space, because our techniques lead to an *over-representation* of the subset. This imposes some extra numerical overhead. In most cases, the quality of the results justifies this additional cost.

3. An Example: Starshaped Zero Sets

In this section, a method to enforce the zero set to be starshaped, introduced in [10], will be described. Recall that a closed curve is starshaped if there is an interior point S from which the whole curve is visible; that is, every ray emanating from S intersects the curve exactly once. Such a point is called a ***kernel point***. For simplicity, we shall assume that this point is the origin; however, it is easy to incorporate into the algorithm a step which will attempt to look for a different kernel point, by simply allowing the fitted polynomial to translate.

As demonstrated in [9], some fits to starshaped objects may have pathologies in them – holes, loops, "folds", extraneous components. Such pathologies can be avoided by forcing the fit to be starshaped.

We force the zero set to be starshaped by allowing every line through a given point to intersect it only twice. We can achieve this in the following way, demonstrated for a quartic polynomial in two variables x and y:

$$P(x,y) = a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00} \quad (1)$$

The value of $P(x,y)$ on a line $y = \alpha x$ through the origin is

$$P_\alpha(x) = (a_{40} + a_{31}\alpha + a_{22}\alpha^2 + a_{13}\alpha^3 + a_{04}\alpha^4)x^4 + (a_{30} + a_{21}\alpha + a_{12}\alpha^2 + a_{03}\alpha^3)x^3 + (a_{20} + a_{11}\alpha + a_{02}\alpha^2)x^2 + (a_{10} + a_{01}\alpha)x + a_{00} \quad (2)$$

If a line through the origin intersects the zero set in more than two points, then $P_\alpha(x)$ will have more than two roots. By applying Roll's theorem twice, it follows that $\frac{d^2}{dx^2}P_\alpha(x)$ should have at least one root. To prevent this, we require that

$\frac{d^2}{dx^2}P_\alpha(x)$ be positive for every x and α. This actually forces the restriction of $P(x, y)$ to every straight line through the origin to be convex (as a function of one variable).

$$\frac{d^2}{dx^2}P_\alpha(x) = 12(a_{40} + a_{31}\alpha + a_{22}\alpha^2 + a_{13}\alpha^3 + a_{04}\alpha^4)x^2 + 6(a_{30} + a_{21}\alpha + a_{12}\alpha^2 + a_{03}\alpha^3)x + 2(a_{20} + a_{11}\alpha + a_{02}\alpha^2) \quad (3)$$

When written in general form (without the scalars resulting from taking derivatives by x), all possible $\frac{d^2}{dx^2}P_\alpha(x)$ are a subset of the set of Sextic polynomials in x and α of the following type:

$$(a_{40} + a_{31}\alpha + a_{22}\alpha^2 + a_{13}\alpha^3 + a_{04}\alpha^4)x^2 + (a_{30} + a_{21}\alpha + a_{12}\alpha^2 + a_{03}\alpha^3)x + (a_{20} + a_{11}\alpha + a_{02}\alpha^2) \quad (4)$$

The challenge is, therefore, to find a parameterization which will cover as many of these polynomials as possible; in particular, we want it to have 15 degrees of freedom. We achieve this by parameterizing polynomials of this type which are everywhere positive.

Denoting this class of polynomials by $\mathcal{POS}_2^4$, we would like to parameterize some *subset* of it. Obviously, we want this subset to be as large as possible, to allow us as much flexibility as possible in the fitting process. The larger the subset, the larger number of shapes which can be described by zero sets of polynomials in it.

We can generate polynomials that are everywhere positive by summing the squares of other polynomials. Thus, a sum of squares of polynomials of the type

$$L_{21}\alpha^2 x + L_{20}\alpha^2 + L_{11}\alpha x + L_{02}x^2 + L_{10}\alpha + L_{01}x + L_{00} \quad (5)$$

is certainly an element of $\mathcal{POS}_2^4$.

The sum

$$\sum_i [L_{21}^{(i)}\alpha^2 x + L_{20}^{(i)}\alpha^2 + L_{11}^{(i)}\alpha x + L_{02}^{(i)}x^2 + L_{10}^{(i)}\alpha + L_{01}^{(i)}x + L_{00}^{(i)}]^2 \quad (6)$$

allows to easily parameterize a subset of $\mathcal{POS}_2^4$. Then, it is straightforward to parameterize polynomials with starshaped zero sets, simply by equating coefficients – lack of space does not allow to present the details.

Denote the polynomials of the type $L_{21}\alpha^2 x + L_{20}\alpha^2 + L_{11}\alpha x + L_{02}x^2 + L_{10}\alpha + L_{01}x + L_{00}$ as $\mathcal{ROOT}_2^4$. Some elements of $\mathcal{POS}_2^4$ are sums of squares of elements of $\mathcal{ROOT}_2^4$. Note that $\mathcal{POS}_2^4$ is a subset of the Sextic polynomials in α and x, and $\mathcal{ROOT}_2^4$ is a subset of the cubic polynomials in α and x.

Finally, denote by $\mathcal{SUMSQ}_2^4$ the subset of the polynomials in $\mathcal{POS}_2^4$ which are sums of squares of polynomials in $\mathcal{ROOT}_2^4$.

Some questions immediately arise:

- Is every element of $\mathcal{POS}_2^4$ a sum of squares of elements of $\mathcal{ROOT}_2^4$? Namely, are the sets $\mathcal{SUMSQ}_2^4$ and $\mathcal{POS}_2^4$ identical?

- If not, does $\mathcal{SUMSQ}_2^4$ have a "full dimension"? That is, what is its dimension (or, equivalently, how many degrees of freedom does it have)? Naturally, we hope that its dimension is 15, as this will guarantee that we are not losing any degrees of freedom.

- What is the minimal number (if it exists at all) of elements of $\mathcal{ROOT}_2^4$ which must be squared and summed in order to obtain all elements of $\mathcal{SUMSQ}_2^4$? This is important when implementing the fitting procedure, for we have to know how $\mathcal{SUMSQ}_2^4$ is to be parameterized. The optimal choice would be to sum as many squares of elements of $\mathcal{ROOT}_2^4$ which will guarantee that we have covered all elements of $\mathcal{SUMSQ}_2^4$. If we sum too many, we are complicating the fitting procedure without gaining anything. If we sum too few, we are losing part of $\mathcal{SUMSQ}_2^4$, and the results of the fitting process will not be optimal.

Next, the answers to these questions – for $\mathcal{SUMSQ}_2^4$ as well as for more general families of polynomials – will be presented, together with some recent results about positive polynomials.

4. Polynomials Represented as Sums of Squares

Most of the material in this section is a short summary of some notions and a few recent powerful results in real algebraic geometry, summarized from [4]. We restrict ourselves to definitions and results which are necessary for the sequel.

First, some terminology:

- Given a ring R, its *Pythagoras number*, $P(R)$, is defined to be the lower bound on the number of squares which must be summed in order to obtain every element of R which is a sum of squares. That is, if any element of R is a sum of squares of elements of R, it can be expressed as a sum of no more than $P(R)$ squares, and $P(R)$ is the minimal number with this property. There is no general bound on $P(R)$, and for some rings it equals infinity; fortunately, that is not the case for the rings of polynomials which are relevant to the fitting paradigm described here.

  The Pythagoras number $P(R)$ is very important for parameterizing the elements of $\mathcal{SUMSQ}_2^4$ (as well as polynomials which are sums of squares of higher degree polynomials). This is because we know that the parameterization given in the previous section requires summing exactly $P(R)$ squares and no more. Naturally, the smaller $P(R)$ is, the better; and, fortunately, some powerful lower bounds for $P(R)$ have been recently obtained for some polynomials rings.

- A *form* is a homogeneous polynomial. The ring of forms of degree m in n variables is denoted $F_{n,m}$, and its Pythagoras number is denoted $P(n, m)$.

- Suppose we are given a subset A of $F_{n,m}$, and let $\mu = ax_1^{i_1}x_2^{i_2}...x_n^{i_n}$ be a monomial which appears in A (hence $\sum_j i_j = m$). The exponent of μ is the vector $(i_1, i_2...i_n)$. Then the *cage* associated with A is the set of all exponents of all the nonzero monomials of A.

  For example, look at the polynomials we have discussed before

$$(a_{40} + a_{31}\alpha + a_{22}\alpha^2 + a_{13}\alpha^3 + a_{04}\alpha^4)x^2 + \\ (a_{30} + a_{21}\alpha + a_{12}\alpha^2 + a_{03}\alpha^3)x + (a_{20} + a_{11}\alpha + a_{02}\alpha^2) \quad (1)$$

  When homogenized, these polynomials assume the shape

$$a_{40}x^2w^4 + a_{31}\alpha x^2w^3 + a_{22}\alpha^2x^2w^2 + a_{13}\alpha^3x^2w \\ +a_{04}\alpha^4x^2 + a_{30}xw^5 + a_{21}\alpha xw^4 + a_{12}\alpha^2xw^3 \\ +a_{03}\alpha^3xw^2 + a_{20}w^6 + a_{11}\alpha w^5 + a_{02}\alpha^2w^4 \quad (2)$$

  And their cage, when viewed as a subset of $F_{3,6}$, is equal to $\{(4,2,0),(3,2,1),(2,2,2),(1,2,3),(0,2,4),(3,1,2),$ $(2,1,3),(1,1,4),(0,1,5),(2,0,4),(1,0,5),(0,0,6)\}$

- For a cage C, let us denote by $l = l(C)$ the number of monomials in C, by $e = e(C)$ the number of even monomials in C (that is, the n-tuples all of whose exponents are even), and by $a = a(C)$ the number of distinct means of even monomials.

  For example, for the cage described above, $l = 12$, $e = 5$, and $a = 12$. This is because every monomial in the cage can be expressed as an average of two even monomials; for instance, $(1,2,3)$ is the average of $(4,2,0)$ and $(0,0,6)$, both of which are even monomials in the cage.

  Let us also denote by $F^+(C)$ the set of everywhere positive polynomials with coefficients in C, and by $F(C)$ those polynomials in $F^+(C)$ which are sums of squares.

  Lastly, let us define the Pythagoras number of a cage C, $P(C)$, in exactly the same fashion as the Pythagoras number of a ring R, that is, as the maximal number of squares that we need to sum in order to obtain all the elements of $F(C)$.

Now, we are ready to present some results from [4]:

Lemma 1 *The dimension of $F^+(C)$ is l, and the dimension of $F(C)$ is a.*

Theorem 1 *For any cage C, the following inequality holds:*

$$\frac{a}{e} \le \lambda \le P(C) \le \Lambda \le e$$

where $\Lambda = \frac{\sqrt{1+8a}-1}{2}$ *and* $\lambda = \frac{2e+1-\sqrt{(2e+1)^2-8a}}{2}$.

Lemma 2 *For any m, $P(3,m) \leq \frac{m}{2}+2$ (this result was obtained by David Leep).*
Lemma 3 *$P(3,4)=3$ (this is a famous theorem of David Hilbert [7]).*
Lemma 4 *For every n, $P(2,n)=2$.*
Lemma 5 *In general, $F^+(C) \neq F(C)$, that is, there are polynomials which are everywhere positive but are not sums of squares.*

Let us see how these results apply to the simplest case we have studied, quartics in two variables:

- The dimension of $\mathcal{SUMSQ}_2^4$ is 12 (Lemma 1). To this, we should add 3 degrees of freedom (because the linear coefficients a_{10}, a_{01}, and the constant coefficient a_{00} are not constrained). Since this gives, altogether, 15 degrees of freedom, we lose no degrees of freedom by using the aforementioned parameterization for quartics in two variables, because they also have 15 degrees of freedom.
- $\mathcal{POS}_2^4 \neq \mathcal{SUMSQ}_2^4$ (Lemma 5). Hence, although we lose no degrees of freedom, there are everywhere positive polynomials which cannot be represented as sums of squares.
- Since $P(3,m) \leq \frac{m}{2}+2$ (Lemma 2), and in our case $m=6$, we need a sum of 5 squares of elements of $\mathcal{ROOT}_2^4$ to guarantee that we have indeed covered all of $\mathcal{SUMSQ}_2^4$.

The first and second observations carry over to higher-degree polynomials and polynomials in three variables, the only change being the upper bound on the Pythagoras number. Note that, because we have to homogenize the polynomials, polynomials in two variables transform into forms in three variables, and polynomials in three variables transform into forms in four variables. For the first, the bound $P(3,m) \leq \frac{m}{2}+2$ is sharper than the one given by Theorem 1. For the latter, we use Theorem 1 to obtain a lower bound; for instance, the lower bound for quartics in 3 variables turns out to be 11. This means that we have to sum 11 squares of polynomials of the appropriate type to guarantee that we obtain all the polynomials which are sums of squares.

4.1. *Some Results*

We have tested the algorithm for fitting starshaped zero sets on a variety of two and three dimensional data sets [10]. To give a flavor of the method's performance, we present two quartic fits to the contour of a human eye (Figure 1). As one can see, the unconstrained fit has spurious components, while the starshaped fit achieves a nice fit. The number of degrees of freedom necessary to parameterize the type of polynomials used was 30.

5. Tight Fits for Convex Planar Polygons

In some applications, such as ray-tracing in computer graphics, and obstacle avoidance in robotics, it is desirable to use a tight fit around convex polyhedral shapes as

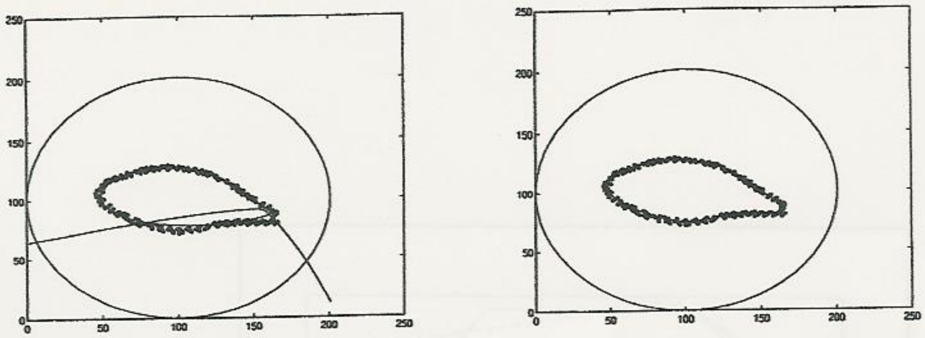


Figure 1: (a) Unconstrained fit to an eye (b) Starshaped fit to an eye

a "bounding volume". If the bounding volume is significantly simpler (in functional form) than the bounded object, it may be used to perform efficient "quick rejects", so that the complex bounded object does not participate in the computation at all. Since evaluating a 2D polynomial of relatively low degree is much simpler that computing the intersection of a ray with a many-sided polygon, substantial computation may be saved. All this, of course, is worthwhile if the bounding volume fits tightly enough, otherwise many "false alarms" result from the extra volume. It is also desirable to restrict the search to a family of polynomials having some "nice" properties. This will ensure that the resulting zero set cannot violate some conditions of topological integrity – for instance, that it cannot have components which are far away from the polygon.

5.1. *A Naive Method*

The naive way to fit implicit polynomials to convex polygons in the plane is to use the paradigm described in the previous section to force the fit to be starshaped. The major problem with this is that the zero set is not guaranteed to properly contain the polygon – which is a necessary requirement in the applications mentioned above. One can try to overcome this by expanding the polygon, and fitting to this expansion. However, it is not clear by what factor to expand the polygon; moreover, it is desirable to have a tight fit around the original polygon, without adding any extra area/volume.

In Figure 2, an example of such a fit is given. As expected, it shows that the family of polynomials described above does not solve the problem of tight fits to convex objects.

In the next section, we introduce a new family of polynomials, which have been quite successful for tight fitting, on the convex polygons we have tested.

5.2. *A Better Method*

In this section, methods to enforce the zero set to be tight around a convex planar polygon will be described. Let us start with a simple example, where the polygon is a square with vertices $(-1,-1),(1,-1),(1,1),(-1,1)$.

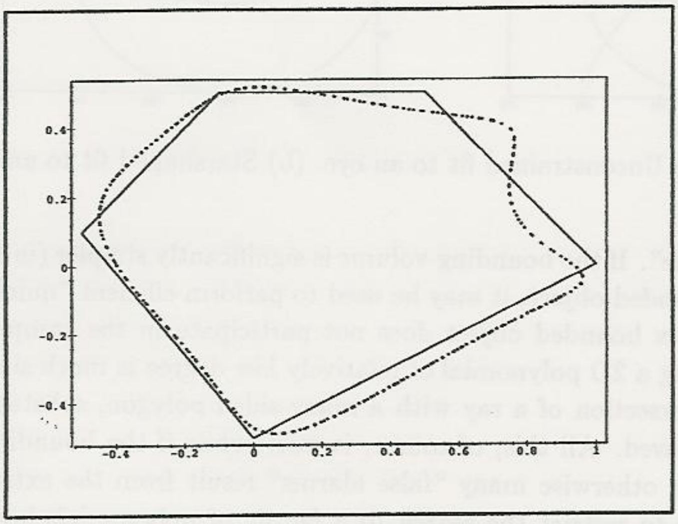


Figure 2: Sextic "starshaped fit" to a convex polygon

Since we are only interested in polynomials whose zero set contains the square, one necessary condition the polynomial has to satisfy is that its sign inside the square will be constant. Without loss of generality, assume that the polynomial has to be negative inside the square. This will guarantee that the containment condition is fulfilled.

However, we also want the fit to be tight. This means that we want the polynomial's sign to change immediately as we move out of the interior of the polygon. For that to happen, we must construct polynomials with a structure that allows them to be positive in any point which is not in the polygon. We describe a simple mechanism to do this, for the case of the square – it generalizes to every convex polyhedral shape in two or three dimensions.

Let $pos_1, pos_2, pos_3, pos_4$ be everywhere positive polynomials in the plane. We have seen how to parameterize such polynomials in Section 4. Consider the following polynomial:

$$POLY_5(x,y) = (-1-x)pos_1(x,y) + (-1-y)pos_2(x,y) + (x-1)pos_3(x,y) + (y-1)pos_4(x,y) \qquad (1)$$

It is easy to see that:

- Each of the four summands compromising $POLY_5$ is negative inside the square; hence, $POLY_5$ will also be negative inside it.

- At every point which is not inside the square, at least one of the summands compromising $POLY_5$ is positive.

So, polynomials such as $POLY_5$ are suitable candidates for tight fits for a square. One can proceed to directly fit the square's boundary with a polynomial of type $POLY_5$, as in [9, 19, 16, 3]. However, this will not guarantee a "nice" fit; e.g. the zero set may have other components near the square. We have used the following method to overcome this problem, not only for the square but for other polygons. Instead of fitting only the polygon itself, we fit the *distance transform* of the polygon. The distance transform is simply the (signed) distance from the polygon's boundary (negative inside the polygon and positive outside).

The major advantage of fitting the distance transform is that it has exactly the properties we want the tight fit function to satisfy: it is zero on the boundary and grows quickly as we move away from it (it is therefore not surprising that it is used as a "potential function" in some obstacle avoidance algorithms [12]). If the fitted polynomial inherits this behavior, it will grow rapidly as we move from the polygon, precluding the possibility of having extraneous components of the zero set away from the polygon.

However, forcing the polynomial to approximate the distance transform over a large region of the plane is an overkill; the family of polynomials has only a limited number of degrees of freedom, and the more constraints we force on it, the more we will have to pay in terms of its tightness around the polygon. We have therefore tried to fit to only a small number of "bands" around the polygon's boundary (by "band" we mean a set of points at roughly fixed distance from the polygon's boundary). This idea was recently applied to general fits [13].

Nonetheless it turns out that it is impossible to avoid extraneous components of the zero set when using polynomials of the type of $POLY_5$ (Eq. (1)). If we choose $pos_1, pos_2, pos_3, pos_4$ to be of degree 4 (which we usually do), $POLY_5$ will be of degree 5. In any case, its degree will be odd, because a sum of squares of polynomials is always of even degree, and we are multiplying these sums by linear terms. This is a serious liability, as the zero set of an odd-degree polynomial is unbounded [9]; therefore, there will always be undesirable components of the zero set away from the polygon.

In Figure 3, the zero set of a polynomial of type $POLY_5$ is shown. It gives a reasonably tight fit around the square. However, it has another unbounded component.

We therefore use a slightly different method, which uses Sextic polynomials. Its advantages are twofold:

- Even-degree polynomials can have zero sets which are bounded [9, 19], hence are natural candidates for tight fitting.

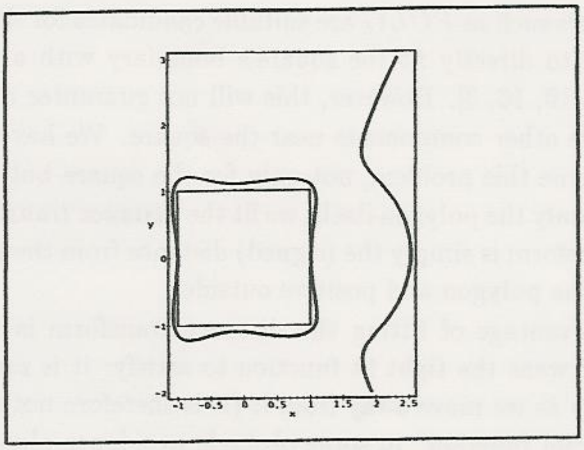


Figure 3: Quintic fit to a square with an extraneous component

- The suggested paradigm for fitting Sextic polynomials requires half the number of free parameters in the fitting than the Quintic method does. For instance, for a square, we need only two positive polynomials to optimize over, instead of four.

Let us now demonstrate how the paradigm works for a square. The idea is essentially the same as for the Quintic fit, however the polynomial has the form

$$POLY_6(x,y) = (x^2-1)pos_1(x,y) + (y^2-1)pos_2(x,y) \tag{2}$$

It is straightforward to see that such polynomials satisfy the necessary conditions (they are negative inside the square and, at any point outside the square, at least one summand is positive). As for the $POLY_5$ shaped polynomials, they can easily be extended to fit any convex polygon. For a square, the result is a very tight fit, shown in Figure 4.

To numerically solve the optimization problem, we have used Lemma 3. It tells us that we need to sum three squares of quadratic polynomials in order to obtain the full range of positive quartics which are representable as sums of squares. Therefore, pos_1 may be parameterized by

$$\begin{gathered}(l_1 x^2 + l_2 xy + l_3 y^2 + l_4 x + l_5 y + l_6)^2 + \\ (l_7 x^2 + l_8 xy + l_9 y^2 + l_{10} x + l_{11} y + l_{12})^2 + \\ (l_{13} x^2 + l_{14} xy + l_{15} y^2 + l_{16} x + l_{17} y + l_{18})^2\end{gathered} \tag{3}$$

and pos_2 by

$$(m_1\, x^2 + m_2\, xy + m_3\, y^2 + m_4\, x + m_5\, y + m_6)^2 +$$
$$(m_7\, x^2 + m_8\, xy + m_9\, y^2 + m_{10}\, x + m_{11}\, y + m_{12})^2 +$$
$$(m_{13}\, x^2 + m_{14}\, xy + m_{15}\, y^2 + m_{16}\, x + m_{17}\, y + m_{18})^2 \qquad (4)$$

Hence $POLY_6$ is dependent on 36 free parameters. In order to obtain the best fit, we minimize the deviation from the distance transform, computed on three bands around the square; it was good enough to take it to be -0.1 on the inner band, 0 on the square itself, and 0.1 on the outer band (Figure 4(a)). A set of 100 points on each band were sampled, evenly spread; denoting this aggregate of points (for the three bands) as $\{(x_i, y_i)\}_{i=0}^{i=300}$, $POLY_6$ was determined by minimizing

$$\sum_{i=0}^{i=300} \left[\frac{POLY_6(x_i, y_i)}{||\nabla POLY_6(x_i, y_i)||} - distance(x_i, y_i) \right]^2 \qquad (5)$$

where $distance(x_i, y_i)$ is the distance transform at (x_i, y_i). The same paradigm

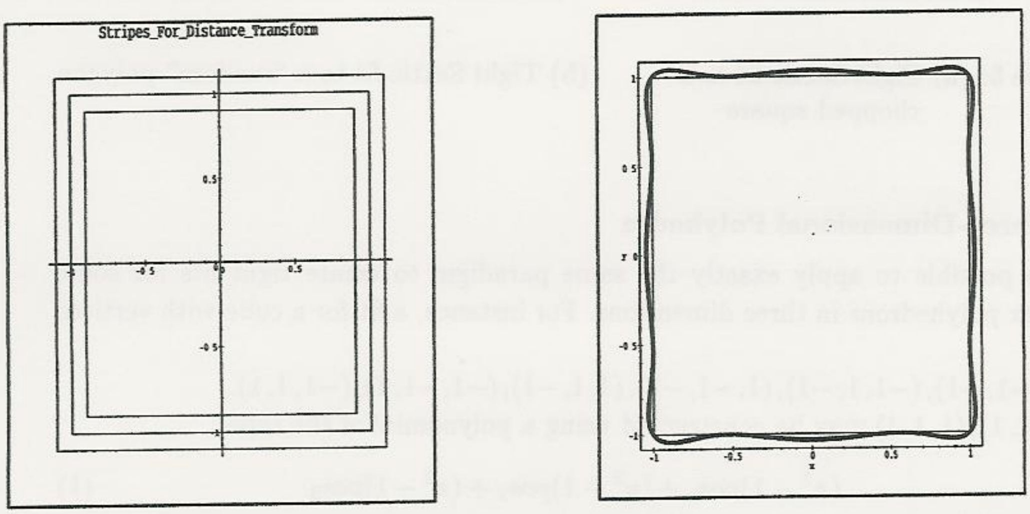


Figure 4: (a) Bands used for distance transform (b) Tight Sextic fit to a square

was applied to fit a chopped square (Figure 5(a)) and a "random polygon" (Figure 5(b)). The latter is the same polygon depicted with its starshaped fit in Figure 2. Note that this polygon has five edges; however, the method can still be applied, by using a polynomial of the shape

$$l_1\, l_2\, pos_1 \;+\; l_3\, l_4\, pos_2 \;+\; l_5\, pos_3 \qquad (6)$$

Where the l_i are linear terms describing the lines which are the extensions of the polygon's edges, and pos_i are everywhere positive polynomials as before. Note that the degree of the polynomial is still six. The optimization problem is similar to that of fitting the square, however there are more degrees of freedom (54) due to the presence of pos_3. The resulting Sextic polynomial will always have 28 coefficients. The running time was a few seconds on a Pentium 100 computer.

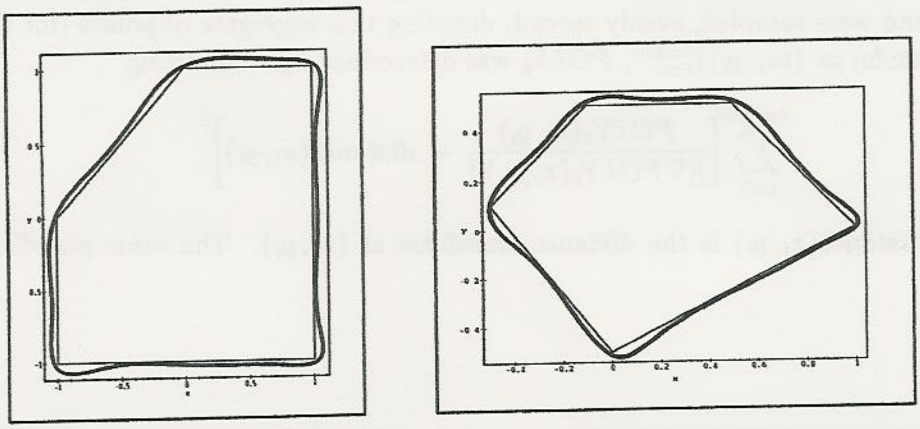


Figure 5: (a) Tight Sextic fit to a chopped square (b) Tight Sextic fit to a "random" polygon

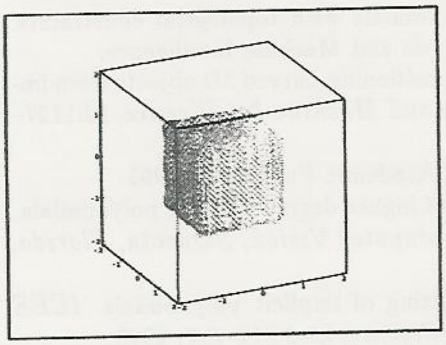
6. Three-Dimensional Polyhedra

It is possible to apply exactly the same paradigm to create tight fits for some convex polyhedrons in three dimensions. For instance, a fit for a cube with vertices at
$(-1,-1,-1), (-1,1,-1), (1,-1,-1), (1,1,-1), (-1,-1,1), (-1,1,1),$
$(1,-1,1), (1,1,1)$ may be constructed using a polynomial of the type

$$(x^2-1)pos_1 + (y^2-1)pos_2 + (z^2-1)pos_3 \qquad (1)$$

where pos_1, pos_2, pos_3 are everywhere positive polynomials in x, y, z. We have also fitted with a Sextic polynomial a "house" shape, consisting of a square and a pyramid on top of it; the results are shown in Figure 6. One can see that they are rather accurate. Note, especially, the pointed vertex of the pyramid; such geometric singularities are a notorious source of problems for fitting implicit models, nonetheless the method suggested here deals with it in a satisfactory manner.

The optimization was carried out as before. However, there are more degrees of freedom – according to Theorem 1, the number of squares that are necessary to represent a positive quartic in three variables is 8 (just substitute $a = 35$ in the theorem, as a general quartic in three variables has 35 coefficients). Since we need three positive polynomials, and since each of the quadratics which has to be summed has six coefficients, the total number of degrees of freedom for the cube fitting is $3 \cdot 8 \cdot 6 = 144$ degrees of freedom. For the "House" example, the number of degrees of freedom was 240, and optimization took about 2 minutes on a Pentium 100 computer. One must remember, though, that the final result is always a Sextic with 84 degrees of freedom; so, although the fitting may be elaborate, it yields a compact result which can be used many times, for a variety of purposes.



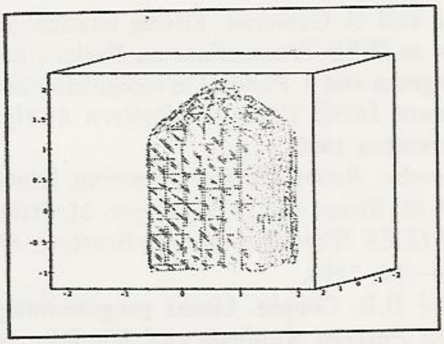


Figure 6: (a) Tight Sextic Fit to a cube (b) Tight Sextic fit for "house"

7. Conclusion

A method for constructing very tight and simple bounding volumes for polyhedra in two and three dimensions was presented. These bounding volumes can potentially be of substantial importance in areas such as graphics and robotics. For starshaped fits, the zero set is guaranteed to be topologically "nice", that is, connected and without folds. For tight fits, we have not been able to prove such properties, however in all the examples we tested the zero set had these desirable properties. The fitted polynomials can also be used to quickly compute an approximation of the Euclidean distance from the bounded polygon, when it is known that we are near the polygon.

1. I. Adler and R. Shamir. A randomization scheme for speeding up algorithms for linear and convex quadratic programming problems with a high constraints-to-variables ratio. *Mathematical Programming*, 61:39–52, 1993.
2. C. Bajaj, J. Chen, and G. Xu. Modeling with cubic a-patches. *ACM Transactions on Graphics*, 14(2):103–133, 1995.
3. C. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Transactions on Graphics*, 12(4):327–347, 1993.

4. M.D. Choi, T.Y. Lam, and B. Reznick. Sums of squares of real polynomials. *Proceedings of Symposia in Pure Mathematics*, 58.2:103–126, 1995.
5. D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3D object recognition and pose. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:971–992, October 1991.
6. D.A. Forsyth. Recognizing algebraic surfaces from their outlines. In *International Conference on Computer Vision*, pages 476–480, Berlin, May 1993.
7. D. Hilbert. Uber die darstellung definiter formen als summe von formen-quadsraten. *Math. Ann.*, 32:342–350, 1888.
8. D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1143–1149, 1994.
9. D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:38–53, January 1994.
10. D. Keren and C. Gotsman. Fitting implicit polynomials with topological constraints. Accepted to IEEE Transactions on Pattern Analysis and Machine Intelligence.
11. D.J. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12:1127–1138, December 1990.
12. J.C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, 1991.
13. Z. Lei, M.M. Blane, and D.B. Cooper. 3L fitting of higher degree implicit polynomials. In *Third IEEE Workshop on Applications of Computer Vision, Sarosota, Florida*, pages 148–153, 1996.
14. Z. Lei and D.B. Cooper. Linear programming fitting of implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20.2:212–217, 1998.
15. D. Levin and E. Nadler. Convexity preserving interpolation by algebraic curves and surfaces. *Numerical Algorithms*, 9:113–139, 1995.
16. 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.
17. J. Subrahmonia, D. Cooper, and D.Keren. Practical reliable bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:505–519, 1996.
18. S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3D object modeling and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:1183–1196, 1994.
19. G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:1115—1138, November 1991.
20. G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16:287–303, 1994.