Direct Least-Squares Fitting of Algebraic Surfaces

Vaughan Pratt
Sun Microsystems Inc.
and
Stanford University
April 30, 1987

Abstract In the course of developing a system for fitting smooth curves to camera input we have developed several direct (i.e. noniterative) methods for fitting a shape (line, circle, conic, cubic, plane, sphere, quadric, etc.) to a set of points, namely exact fit, simple fit, spherical fit, and blend fit. These methods are all dimension-independent, being just as suitable for 3D surfaces as for the 2D curves they were originally developed for.

Exact fit generalizes to arbitrary shapes (in the sense of the term defined in this paper) the well-known determinant method for planar exact fit. Simple fit is a naive reduction of the general overconstrained case to the exact case. Spherical fit takes advantage of a special property of circles and spheres that permits robust fitting; no prior direct circle fitters have been as robust, and there have been no previous sphere fitters. Blend fit finds the best fit to a set of points of a useful generalization of Middleditch-Sears blending curves and surfaces, via a nonpolynomial generalization of planar fit.

These methods all require $(am + bn)n^2$ operations for fitting a surface of order n to m points, with $a = 2$ and $b = 1/3$ typically, except for spherical fit where b is larger due to the need to extract eigenvectors. All these methods save simple fit achieve a robustness previously attained by direct algorithms only for fitting planes. All admit incremental batched addition and deletion of points at cost an^2 per point and bn^3 per batch.

1. Introduction

Background. We began this project with the problem of recovering outline fonts from scanned-in camera images of large-scale drawings, as part of a larger project to automate the entry of such drawings. Such entry is usually done by hand, typically by entering points around the curve via a tablet and postediting the result to achieve smoothness and fidelity to the original. Such manual entry of curves is not only an expensive use of human resources but also less accurate than what can be achieved by working algorithmically with scanned-in camera images.

The techniques we developed for this application are of general interest to other domains where curve fitting is needed, as well as to situations involving surface fitting in three or more dimensions. This paper therefore emphasizes these general techniques at the expense of their motivating application, which we hope to report on elsewhere.

Our application is also the main application of similar work reported by Plass and Stone in this forum [21]. The two main differences of our work from theirs are our emphasis on algebraic as opposed to parametric curves (in the 2D case), and on one-step application of matrix-inversion-style least-squares methods, which we refer to as *direct* methods in distinction to slower iterative methods.

Problem Statement. We wish to fit an algebraic curve or surface of a given shape to m points in d-dimensional space $\mathbf{R}^d$. We cater for such shapes as circle, line-pair, cubic, cone,

sphere, and quadric, all of which are definable in the form $f(x_1, \ldots, x_d) = 0$ where f is a polynomial in its arguments, but not, e.g., for space curves, which are defined as the intersection of two surfaces and require two such equations.

Goodness of fit is taken to be the length of the m-vector of distances of the m points from the surface. The fit is called exact, best, or good when this length is respectively zero, least possible, or close to least possible.

We take the length of an m-vector to mean its Euclidean length. Best fit then amounts to least-squared-distance fit.

We take the *geometric* distance of a point p from a surface S to be the distance from p to the nearest point of S, i.e. the minimum, over all points p' of S, of the Euclidean distance from p to p'. Unfortunately geometric distance is neither computationally nor algebraically convenient, particularly for higher-order surfaces. This leads to the use of distance metrics that approximate geometric distance. A distance metric is more or less *robust* when the best fit under it is a more or less good fit under geometric distance.

Performance. The main emphasis of this paper is on fast computation of good fits, to which end we confine ourselves to *direct* methods, by which we mean those least-squares methods that require roughly the work of a matrix inversion or an extraction of eigenvalues. Our typical cost to fit a surface of order n to m points is on the order of $(m+n)n^2$ operations, e.g. to fit circles to 20 points $m = 20$ and $n = 4$, so a few hundred operations. By contrast iterative methods may be much slower; Figure 2 of [21] suggests that an iterative method may require on the order of a hundred iterations to fit a parametric cubic to a set of $m = 20$ points, with each iteration itself requiring the application of a least squares algorithm.

Why algebraic surfaces? An algebraic presentation of a surface is an implicit presentation $f(x_1, \ldots, x_d) = 0$ for which f is a polynomial in its d arguments. The main rationale for our interest in algebraically presented surfaces is that they lend themselves to direct least-squares techniques at least as naturally as parametric surfaces. In addition, in some situations parameters constitute an inconvenient or unnatural artifact. Thirdly, although all algebraic surfaces are representable parametrically they are not all representable with a polynomial or even rational polynomial representation, whence algebraic methods have a greater domain of applicability. And fourthly, parametric curves and surfaces have received the lion's share of attention in the fitting literature, creating the misleading impression that algebraically presented curves and surfaces are less suitable for fitting purposes. Indeed our own experience has been that the situation is quite the opposite: we find the algebraic form more convenient and efficient for fitting.

Related work. There appears to be relatively little written about fitting algebraic curves. A fairly thorough search turned up only a few treatments of least-squares fitting of algebraic curves [1, 2, 3, 4, 9, 15, 16, 24, 25] and none whatsoever of least-squares fitting of nonplanar algebraic surfaces. By comparison least-squares fitting of parametric polynomial curves and surfaces is routinely treated in many papers and a number of textbooks [5, 7, 8, 13, 19]. In the case of least-squares fitting of surfaces there seems to be a universal impression that fitting is only feasible for parametrically presented surfaces.

Perhaps the single commonest failing of those papers that do treat algebraically presented curves is their casual adoption of computationally convenient distance metrics. These metrics generally measure the distance of a point (x_0, y_0) to a curve $q(x, y) = 0$ ($q(x, y)$ a polynomial in x and y) by first normalizing q according to some quadratic constraint on its coefficients and then taking the distance to be $q(x_0, y_0)$. We shall call this a *quadratic-norm* distance metric. Bookstein [3] faults several authors [1,2,4,9] for adopting quadratic-norm metrics that depend on choice of basis, and gives a similarity-invariant metric (relative to geometric distance) that is usable for both conic-fitting and circle-fitting, arguing that this is the only similarity-invariant exact-fit-preserving quadratic-norm metric.

Sampson [24] points out that even with these properties Bookstein's metric still departs sufficiently from geometric distance that when fitting highly elliptic conics to "very scattered" data the resulting fit can be perceptibly inferior. One might conclude that the application of Bookstein distance to circle-fitting would be less problematic since the ellipticity problem addressed by Sampson cannot arise. Actually it is the other way around: for circle fitting Bookstein's measure can fail far more convincingly than envisaged by Sampson, which we illustrate below with an example where curvature of the fitted circle increases when it should be decreasing.

Turner [25] and Sampson [24] apply the nonalgebraic distance metric $q/|\nabla q|$ to curve fitting. We show how to modify this metric to be algebraic – a quadratic normalization – for the case of circles (extending immediately to spheres), ironically the one case of ellipses to which Sampson did not evisage applying $q/|\nabla q|$.

Results. The techniques we have developed are as follows.

(i) Exact fit. We give a simple method for exact fitting of a surface of a given shape to the appropriate number of points, generalizing the well-known determinant method for fitting a hyperplane in d dimensions to d points.

(ii) Simple fit. We naively translate the problem of finding the best fitting surface for a surfeit of points into the exact-fit problem, via normal equations. The translation is very simple, has excellent performance, but lacks robustness. We give a variation on the method that substantially improves its robustness at negligible performance cost.

(iii) Spherical fit. We give a distance metric that leads to a robust direct algorithm for fitting spheres in d dimensions. For $d = 2$ this algorithm is the first direct least-squares circle fitter to achieve this level of robustness. For higher d this is still true, albeit vacuously since the problem of fitting spheres to points seems not to have previously been considered.

(iv) Blend fit. We first give a useful generalization of the Middleditch-Sears [14] approach to blending surfaces. Then, given a set of base surfaces to be blended, along with a set of points, we give a simple direct method for finding a blending surface of a given shape best fitting those points.

2. Samples, Surfaces, and Shapes

Given a *sample* consisting of an m-tuple of points in $\mathbf{R}^d$, we wish to find a *surface* of the form $Z(q) \subseteq \mathbf{R}^d$ consisting of the zeros of a function $q : \mathbf{R}^d \to \mathbf{R}$, that comes close to

minimizing the sample-to-surface distance, which we define as the Euclidean norm (length) of the m-vector of the true (geometric) point-to-surface distances. The function q is to be drawn from a given set Q; for example if we are fitting circles then Q is the set Q_C of all polynomials of the form $A(x^2+y^2)+Dx+Ey+F$.

The techniques of this paper apply to sets of functions closed under linear combinations. For example it should be apparent that Q_C is so closed. Such a subset of the ring of all functions $q : \mathbf{R}^d \to \mathbf{R}$, is called an *ideal* of that ring; it also constitutes a vector space ($\mathbf{R}$ being a field). For example Q_C is a 4-dimensional vector space (of polynomials $q : \mathbf{R}^2 \to \mathbf{R}$) one of whose bases is $x^2+y^2, x, y, 1$. (The reader accustomed to thinking about the set of cubic polynomials as a four-dimensional vector space with $1, t, t^2, t^3$ as one basis and $(1-t)^3, 3t(1-t)^2, 3t^2(1-t), t^3$ as another should have no difficulty adapting to these concepts.) Our techniques are limited to the case where this space is of finite dimension n. We call a set of surfaces defined by an n-dimensional ideal a *shape* of order n, e.g. the set of all 2D circles constitutes a shape of order 4.

It is customary in geometry to work with just the ideals of the ring of all *polynomials* $q : \mathbf{R}^d \to \mathbf{R}$. However our methods extend immediately to the larger ring of all *functions* $f : \mathbf{R}^d \to \mathbf{R}$, which we take advantage of for blend fitting.

Every such Q contains the identically zero polynomial 0, defining the trivial surface consisting of the whole space. One additional requirement when fitting surfaces is that we do not allow this trivial fit. We nevertheless leave 0 in Q for algebraic convenience.

Examples. Many useful shapes are definable by finite-dimensional ideals. We list a few below along with suitable generators for their defining ideals, each set of generators constituting a basis for the ideal as a vector space.

horizontal line	$1, y$
diagonal line	$1, x+y$
line*	$1, x, y$
upright parabola	$1, x, y, x^2$
upright hyperbola	$1, x, y, xy$
diagonal hyperbola	$1, x, y, x^2-y^2$
circle at origin	$1, x^2+y^2$
circle	$1, x, y, x^2+y^2$
right hyperbola	$1, x, y, 2xy, x^2-y^2$
conic*	$1, x, y, x^2, xy, y^2$
	or circle $\cup$ right hyperbola
cubic*	conic $\cup\ x^3, x^2y, xy^2, y^3$
plane*	$1, x, y, z$
z-axis cylinder	$1, x^2+y^2$
z-axis cone	x^2+y^2, z^2
x=z-axis cone	$y^2, 2xz$
z-aligned cylinder	$1, x, y, x^2+y^2$
sphere	$1, x, y, z, x^2+y^2+z^2$
right hyperboloid	$1, x, y, z, x^2-y^2, y^2-z^2,$

	$2xy, 2yz, 2zx$
quadric*	$1, x, y, z, x^2, y^2, z^2, xy, yz, zx$
	or sphere $\cup$ right hyperboloid

Asterisks denote shapes invariant under linear transformations (and hence under change of basis), meaning that a linear transformation maps any object of that shape to an object of the same shape. Although the circle, right hyperbola (orthogonal asymptotes, useful for converting spherical fit to conic fit), sphere, and right hyperboloid (3D analog of the right hyperbola) shapes are not so invariant, they are invariant under similarities (angle-preserving linear transformations, i.e. rotations, scalings, and translations), while the properties of being a line of a given orientation (horizontal, diagonal, etc.), upright parabola, upright or diagonal hyperbola (asymptotes parallel to or at 45 degrees to the axes), or z-aligned cylinder are invariant under translations. The remaining properties are not even invariant under translations.

3. Algebraic Distance

A commonly used surrogate for geometric distance from a point p to a surface $Z(q)$ is the value of q at p. Since $Z(q) = Z(cq)$ for $c \neq 0$ q is first normalized in order to make this value meaningful, typically by scaling it so as to set to a constant (unity for definiteness) some quadratic function of its coefficients, which we call *quadratic normalization*. Since we seek to minimize the sum of the *squares* of the distances it changes nothing if we take the negation of a normal polynomial to also be normal. We shall refer to distance computed in this manner as *algebraic distance*; the essential characteristic of algebraic distance from p to $Z(q)$ is that it is computed by evaluating a fixed representative polynomial cq, *chosen independently of* p.

An important aspect of quadratic normalization for our purposes is that the best fit under such a distance metric can be computed directly via the computation of an $n \times n$ eigenvector in $O(n^3)$ operations [12]. Hence *any distance metric defined by a quadratic normalization leads automatically to a fitting method meeting our performance requirement.* The remaining concern is then with the quality of fit, which can vary substantially between normalizations.

Normalization can be visualized geometrically by thinking of the set $\{cq|c \text{ real}\}$ for any given q as the line containing the polynomials q and 0 in the vector space Q. This line is called the *principal ideal* generated by q, and pervades the algebraic geometry of surfaces. Normal polynomials then appear as complementary pairs of points (equidistant from the origin of Q) on principal ideals, and we may think of the set of all normal polynomials as forming a surface in the space Q, which we may call a *normalizing surface* in Q (not to be confused with surfaces comprising shapes, which exist in R^d).

If the normalizing surface is say a sphere then it will intersect all principal ideals (in two complementary points), but if it is say a cylinder then the principal ideal along its axis will contain no normal polynomial (equivalently, the normal polynomial can be regarded as being at the "end" of the ideal, i.e. at infinity). In this case a fitting algorithm will never fit such a principal ideal; furthermore, principal ideals very close to it will have very large normal polynomials and so will appear to be bad fits. We think of the unfittable

polynomials as the *singularities* of the normalization, and their neighbors as being very hard to fit. In the actual fitting process the presence of singularities is felt as a sort of repulsive force pushing the fitted shape well away from the singularity.

A number of authors have proposed such normalizations for the conic shape. Paton [16] normalizes conics $Ax^2+Bxy+Cy^2+Dx+Ey+F=0$ subject to $A^2+B^2+C^2+D^2+E^2+F^2=1$, corresponding to taking the normalizing surface to be the unit sphere with center 0 (with respect to the basis $x^2, xy, y^2, x, y, 1$), having no singularities. Biggerstaff [2], Albano [1], and Cooper and Yalabik [4] take the plane $F=1$ (equivalently, the two planes $F=\pm 1$), whose singularities correspond to those conics that pass through the origin. Gnanadesikan [9] takes the unit cylinder along the F-axis, that is, $A^2+B^2+C^2+D^2+E^2=1$, missing only the F-axis itself, defining the empty conic $Z(1)$, which is no great loss. Bookstein uses the ellipsoidal cylinder $A^2+B^2/2+C^2=2$, ruling out the subspace $A=B=C=0$ of Q which can be seen to make straight lines singular and so unfittable.

Each of these metrics save Bookstein's varies (relative to geometric distance) under similarities (rotations, translations, and scaling) of the plane, as Bookstein points out. It is worth adding that the most popular normalization, $F=1$, is a particularly poor one due to the singularity at the origin, which tends to push the fit away from the origin. (Hence to lie with statistics when fitting curves to data it suffices to choose as the origin for that data a point you want the curve to stay away from regardless of what the data says.) Gnanadesikan's normalization has the opposite problem, tending to push the fit towards the origin to keep D and E small.

For the circle shape, definable as the subshape $A=C, B=0$ of the conic shape, Bookstein's normalization specializes to $A^2+C^2=2$, i.e. $A=\pm 1$, still having lines as its singularities. One might then predict that Bookstein's normalization should prefer slightly more curved fits than the true best fit; in actuality it is easily encouraged to prefer absurdly curved fits, as we shall see in the section on spherical fits. To correct this we propose a new normalization for circles, namely $D^2+E^2-4AF=1$ ($D^2+E^2+F^2-4AG=1$ for spheres). Like Bookstein's normalization this is invariant relative to geometric distance under similarities but has only points (zero-radius circles) as singularities, which appear to cause less havoc than lines.

4. Nonalgebraic Distance

We mention here some of the principles behind metrics whose normalizations do depend on p, i.e. nonalgebraic distance metrics. Curiously enough, one of the insights from this section leads us to our above-mentioned algebraic distance metric for circles. Beyond this, an understanding of the principles will improve perspective on surface fitting techniques in general.

Sink or Swim. In visualizing the correspondence between geometric and algebraic distance we find the *sink or swim* picture helpful. In dimension $d=2$, think of a point $p=(x,y)$ on the plane R^2 as a swimmer in the ocean and the 3D surface $z=q(x,y)$ as the land below him. $Z(q)$ is the shoreline, geometric distance $D(p,Z(q))$ is swimming distance to shore, unnormalized algebraic distance (the sign is unimportant) is sinking distance to bottom, and normalization is vertical rescaling. In this connection a useful additional concept is

the gradient operator ∇: $\nabla q = (\partial q/\partial x, \partial q/\partial y)$ is a function which assigns to each point p on the ocean surface a vector $\nabla q(p)$ lying in the ocean surface whose direction is the uphill direction of the ocean bottom immediately below p and whose length $|\nabla q(p)|$ is the slope of the ocean bottom there.

When the surface $z = q(x, y)$ is planar, ∇q and hence $|\nabla q|$ are constant. We can therefore normalize q to $q/|\nabla q|$ to yield a surface with slope 1, for which algebraic distance coincides with geometric distance [18]. If $q(x, y) = Dx + Ey + F$ then $|\nabla q| = \sqrt{D^2 + E^2}$, so the appropriate normalization is $D^2 + E^2 = 1$, with only the trivial degeneracy $F = 0$. This leads to an eigenvector-based direct method for best geometric fitting of lines and planes in any R^d. (A slope-1 normalization is possible also for cones, pyramids, and much more complex surfaces; unfortunately none of these shapes appear to be definable in terms of an ideal of polynomials. They can in some cases be defined in terms of an ideal of algebraic functions, typically involving square roots of polynomials, but unlike the convenient setup with blending surfaces that we present below these ideals appear not to be of finite dimension, ruling out any direct application of our methods.)

Turner [25] and Sampson [24] have independently proposed using the above normalization $q/|\nabla q|$ for nonplanar shapes, for which $|\nabla q|$ is not constant. This normalization is a function of p and gives a nonalgebraic distance metric, albeit one that remains computationally more tractable than geometric distance. Nalwa and Pauchon [15] refine this metric to take into account second-order derivatives of q, which can be helpful with very scattered data.

These metrics offer the following benefits. First $q/|\nabla q|$ is insensitive to scaling of q. Secondly it is as invariant as geometric distance, being invariant under translations and rotations and varying in proportion to geometric distance under change of scale; hence the best fit is invariant under similarities (angle-preserving transformations or changes of basis). Thirdly it coincides with geometric distance for plane surfaces. Fourthly, for nonplanar surfaces, $q(p)/|\nabla q(p)|$ approximates geometric distance to the extent that q is approximately planar (i.e. approximates a linear combination of $1, x_1, \ldots, x_d$) on the (d-dimensional) ball with center p and radius $q(p)/|\nabla q(p)|$, which is almost invariably the case for only slightly scattered data.

The Turner-Sampson and Nalwa-Pauchon metrics are both nonalgebraic and seemingly unusable with direct methods. Rather, at least as envisaged by Sampson [24], one iteratively computes an algebraic fit q by a direct method, weighting the algebraic distance from each sample point p to $Z(q)$ by $1/|\nabla q^{(-1)}(p)|$ where $q^{(-1)}$ is the surface found at the previous iteration, using unit weights in the first iteration. This appears to us to be the most robust method for those situations where there is no appropriate quadratic normalization, e.g. highly elliptical conics, as we discuss in the section on spherical fits.

5. Exact Fit

We give a straightforward method for exactly fitting a surface of order n to $n - 1$ points $p_1, \ldots, p_{n-1}$, which we want mainly for the more general problem of approximately fitting such a shape to at least that many points. The method is well-known for the case of planes, appears occasionally in textbooks for the case of circles, and in [22] and [20] (p.369) for conics. However we have been unable to locate any reference to the general method.

Let $A : Q \to \mathbf{R}^m$ map each polynomial $q \in Q$ to the m-vector of values of q at the m points, evidently a linear transformation. The exact fits are then the zeros of A. Given any basis $b_1, \ldots, b_n$ for Q, q is representable as the n-vector $\mathbf{q}$ of coefficients of b_i's A is representable as the $m \times n$ matrix A whose ij-th element is $b_j(p_i)$, and $A(q)$ is given by the product $A\mathbf{q}$. To fit a circle to five points we would have

$$A = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 + y_1^2 \\ 1 & x_2 & y_2 & x_2^2 + y_2^2 \\ 1 & x_3 & y_3 & x_3^2 + y_3^2 \\ 1 & x_4 & y_4 & x_4^2 + y_4^2 \\ 1 & x_5 & y_5 & x_5^2 + y_5^2 \end{pmatrix}.$$

Then the matrix A amounts to a change of coordinates for (defining polynomials of) surfaces, namely from the given basis to a coordinate system in which the i-th coordinate gives the algebraic distance of the surface from p_i.

For the case $m = n - 1$ an exact fit is possible, and is easily found as follows. Let A^+ denote the square matrix obtained from A by adjoining an n-th row consisting of the basis polynomials themselves, making A^+ a matrix over polynomials, and form its determinant. In the circle example this determinant is

$$\begin{vmatrix} 1 & x_1 & y_1 & x_1^2 + y_1^2 \\ 1 & x_2 & y_2 & x_2^2 + y_2^2 \\ 1 & x_3 & y_3 & x_3^2 + y_3^2 \\ 1 & x & y & x^2 + y^2 \end{vmatrix}$$

This determinant, a polynomial q, can be seen to be a linear combination of the n (here $n = 4$) basis polynomials and so is in Q, whence we have a legal surface. Since the value of row n at p_i is row i it follows that the determinant vanishes at each p_i, so the surface passes through all $m = n - 1$ points.

We may compute q as the cofactors of the elements of the n-th row, giving its representation in terms of coefficients of the $b_i's$. For large n it is worthwhile to triangularize A first (i.e. make $A_{ij} = 0$ for $i > j$ via row operations) at cost $\mathrm{O}(n^3)$ and then compute the cofactors at an additional cost of $\mathrm{O}(n^2)$.

The one uninteresting case of this situation is when q is identically 0, which it is if and only if the rank of A is strictly less than $n - 1$. In this case the points underdetermine the shape, a situation we do not treat here.

When $m \geq n$ but the rank of A is $n - 1$ we may select $n - 1$ linearly independent rows of A to form an $(n - 1) \times m$ matrix whose rank is still $n - 1$. The above technique may then be applied to this matrix to yield a surface passing through these $n - 1$ selected points. This surface will also pass through all points whose corresponding matrix rows are linear

combinations of the $n-1$ selected rows, which is the case for the $m-(n-1)$ unselected rows, the rank of A being only $n-1$. Again, if the rank of A is less than $n-1$ the points still underdetermine the shape.

6. Simple Fit

The only remaining case now is $m \geq n$ and $\text{rank}(A) = n$. This is the overdetermined case (no exact fit), our primary interest. The goal is to find a good fit $Z(q)$ to the sample. The following method is of interest partly for its simplicity, partly for its connection to Exact Fit, and partly for how it circumvents singularities.

We first state the method for the normalization which holds the last coefficient constant. This normalization has the obvious drawback that a fit having this coefficient zero constitutes a singularity, which we will attend to shortly.

The first step of the method is the basic step for the method of normal equations, the second step is novel in that the normal equations method is usually applied to systems with an independent variable, whereas here we are solving an implicit system in which none of the d variables can be identified as independent.

The Simple Fit algorithm can be obtained by combining the non-geometric ideas from equation (19.10) of [12] (taking their $\tilde{U}$ to be our U) with the above geometrically-oriented exact fit algorithm, as follows.

1. Given A as above, of size $m \times n$, compute the Cholesky decomposition $A'A = U'U$ [7,8,12,19]. That is, compute the unique $n \times n$ upper triangular U with nonnegative diagonal entries such that $U'U = A'A$.

2. Delete the last row of U to yield an $(n-1) \times n$ matrix and treat the result as though it were the $(n-1) \times n$ matrix A in the exact fit case. That is, append a row of polynomials and form the determinant.

The discussion of equation 19.10 in [12] shows that the resulting surface is the best algebraic fit subject to holding the n-th coefficient of q constant. (Hence the best algebraic fit under the normalization in which the n-th coefficient is 1 may be obtained by dividing q by its n-th coefficient.) The quality of fit (square root of sum of squares, amounting to standard deviation times $\sqrt{n}$) is given by U_{nn} (ρ in 19.10 [12]), the one nonzero element of the discarded last row.

Cholesky Without Square Roots. While the $U'U$ decomposition has the merit of conceptual simplicity it has the drawback of requiring the extraction of n square roots. Cholesky decomposition without square roots (Exercise 19.40 of [12]) modifies step 1 above by finding $n \times n$ matrices U and D satisfying $A'A = U'DU$ where D is diagonal and the leading diagonal of U consists of 1's. Step 2 is left unchanged.

A suitable procedure for this decomposition is as follows. An $n \times n$ matrix P is initialized to $A'A$. Only the upper triangles of P and U require storage since $A'A$ is symmetric and

U is upper triangular.

for $i := 1$ **to** n **do**

$\{U_{ii} := 1;$

for $j := i + 1$ **to** n **do**

$\{U_{ij} := P_{ij}/P_{ii};$

for $k := j$ **to** n **do**

$P_{jk} := P_{jk} - U_{ij} \times P_{ik}\}\}$

Note that the procedure modifies P. D is obtained as the diagonal of the final P. If $P_{ii} \leq 0$ (with negative P_{ii} being possible only on account of roundoff error) then U_{ij} is set to 0 for all $j \geq i$. The diagonal of U being 1 simplifies the computation of the cofactors in Step 2, which requires only $O(n^2)$ operations.

The quality of fit is now obtained as D_{nn} rather than U_{nn}, the latter now always being 1. D_{nn} is actually the square of the old U_{nn} and so is the sum of squares rather than its square root (amounting to variance times n). The rest of D may be discarded since q and hence each row of U need only be determined up to a constant factor. Besides avoidance of square roots this decomposition has the property that the last coefficient of q produced in Step 2 is 1; the resulting q is as for the $U'U$ decomposition after division by its n-th coefficient. (In this respect the method acts as though the n-th coordinate were the independent variable in a conventional least-squares regression.) Hence for conics the popular $F = 1$ normalization can be implemented with this method by putting 1 at the end of the basis, and for circles Bookstein's normalization $A = 1$ can be used by putting $x^2 + y^2$ at the end, when these normalizations are appropriate.

The principle novelty in the above is the application of well-understood least squares techniques, using normal equations and Cholesky decomposition, to fitting algebraic surfaces.

Basis Order Independence. Ideally a procedure for selecting a member of Q would be independent of the choice of basis for Q. This is possible using a somewhat more elaborate procedure than we shall consider here. With considerably less effort we are able to achieve independence of the order in which the elements of the basis are presented, via a procedure we shall now describe. A corollary of this property is that no one coefficient is singled out as having to be nonzero, eliminating this source of singularities from this application of Cholesky decomposition.

It would be particularly convenient if the algorithm were to hold constant (namely 1) the coefficient with maximum absolute value. Unfortunately this is not the case for the procedure we shall give – we have seen coefficients as large as $n - 1$ (n the size of the matrix). It is tempting to conjecture that this is the limit on size of coefficients. We do not understand at all the mechanism by which the algorithm selects which coefficient is to be held constant.

The idea is to perform row-and-column permutations of the (more or less) as-yet unprocessed P during the Cholesky decomposition. Just before the assignment of 1 to U_{ii}, the

maximum P_{jj} for $j \geq i$ is found, and if $j \neq i$ a row-and-column permutation of P is performed in place, exchanging i with j; in effect the i-th and j-th basis elements are exchanged.

We omit the proof that the result is independent of basis order.

The method lends itself to partial permutations, in which some elements of the basis are not permuted. In fitting lines and planes for example the constant basis element (1) can be left undisturbed if it is put at the beginning of the basis, though we have not encountered a situation where it is actually beneficial to leave it alone.

Incremental addition and deletion of points. The matrix $A'A$ is $n \times n$, which is considerably smaller than A when $m >> n$. Yet despite the extent of the data reduction implied by this compaction it is very easy to update $A'A$ to reflect the addition *or* deletion of points. Each point, as a row of A, forms a $1 \times n$ matrix Z, with $Z'Z$ the same size as $A'A$. To add or delete point Z from A, add or subtract $Z'Z$ from $A'A$. In our implementation we compute $A'A$ exclusively by this method.

Weighting. To increase or decrease the contribution of a point, scale either Z or $Z'Z$ (as convenient) appropriately. Doubling $Z'Z$ has the same effect as having two occurrences of the point. One might decrease the weight of a point if it is relatively unreliable; conversely one might increase its weight to force the surface to pass closer to it.

Cost. If $A'A$ is maintained incrementally the cost to add or delete a point is n^2 multiplications (to form $Z'Z$) and n^2 additions (to add or subtract it). Hence adding or deleting m points in a batch (without then running Cholesky) costs mn^2 such operations. The constant factor in the cost of Choleski decomposition of $A'A$ makes it quite cheap in comparison with Gaussian elimination; the procedure requires only $n^3/6$ multiplications and additions. Because U is triangular computing the determinant of the modified U requires only n^2 multiplications and additions. Thus for circles, with $n = 4$, the cost is $16m + 26$ multiplications and additions. For conics, with $n = 6$, the cost rises to $36m + 36$. The additional cost of the row-and-column-permuting variant is $O(n^2)$ exchanges, which is dominated by the other costs.

Quality of Fit. We have not been able to analyze this method directly. Experience with its use however demonstrates the need for the basis-permuting variant, in the absence of which the singularities consisting of fits with zero n-th coefficient are very noticeable. Permuting the basis eliminates those singularities, but we have noticed in the case of planar fits a tendency to avoid exact 45-degree fits when the data is very badly scattered. It would be of considerable interest to know whether this situation could be understood in terms of the shape of a normalization surface associated with the algorithm.

Stability. In the method of normal equations, all steps save the computation of $A'A$ are numerically very stable; the replacement of A by $A'A$ has the destabilizing effect of squaring the condition number. When A is ill-conditioned, such as when sampling points from two nearly parallel coplanar lines to determine their common plane, normal equations aggravate the situation. This effect may be offset by either (i) doubling precision, (ii) using an alternative method based on Householder or Givens transformations of A [12],

or (iii) designing the application to avoid geometric instabilities. Our preference has been a combination of (i) and (iii), (ii) having somewhat inferior performance to Cholesky decomposition, and considerably worse performance when points are to be added and deleted incrementally, which our application makes extensive use of.

7. Spherical Fit

In this section we give a quadratic-norm metric for circles and spheres which is substantially more robust than the only other such metric to have previously been proposed for circles, namely Bookstein's. As pointed out in the section on algebraic distance, the singularities for Bookstein's metric are lines. Such singularities tend to increase the curvature of the fit. The following illustration of this nonrobustness of Bookstein's metric should give some idea of the rate at which the fit deteriorates as the curve of best fit approaches a line.

Figure 1 shows, under each of the $A = 1$ and geometric distance metrics, the best fitting circle to the points $(-1, 0)$, $(-.3, y)$, $(.3, .1)$, $(1, 0)$ for $y = .1, .02, -.02, -.06$. The $A = 1$ circle is in each case the one with higher curvature, with equality only at $y = .1$.

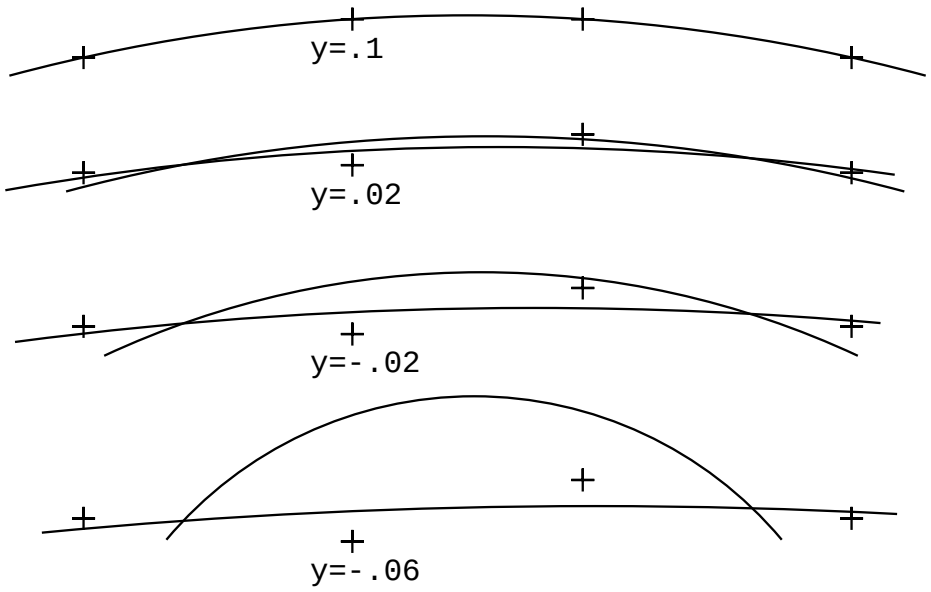

Figure 1. Best fitting circles under $A = 1$ and geometric metrics.

The $A = 1$ circle can be seen to become a very poor fit as the best geometric fit increases in radius.

We repair this problem as follows. The circle ideal consists of all polynomials $q(x, y) = A(x^2 + y^2) + Dx + Ey + F$. For any such q the 3D surface $z = q(x, y)$ (z being the direction of sinking in the sink-or-swim model of Section 4) is a paraboloid of revolution. Our observation is that although $|\nabla q|$ is not constant, it *is* constant on $Z(q)$, by the circular symmetry of the paraboloid. This motivates normalizing q to make $|\nabla q| = 1$ on $Z(q)$.

Now the partial derivatives of q are $\partial q/\partial x = 2Ax+D$ and $\partial q/\partial y = 2Ay+E$. Hence $|\nabla q|^2 = 4A^2x^2 + 4DAx + D^2 + 4A^2y^2 + 4EAy + E^2 = 4A(Dx + Ey + A(x^2 + y^2)) + D^2 + E^2 = 4A(q(x, y)-F)+D^2+E^2$. The term $q(x, y)$ vanishes on $Z(q)$, where $|\nabla q|^2 = D^2+E^2-4AF$. Hence we obtain the invariant $D^2 + E^2 - 4AF = 1$. (The normalization itself, meaning the quantity by which q must be divided to be normalized, is $\sqrt{D^2 + E^2 - 4AF}$. However we need not actually perform this operation; for finding the circle of best fit it suffices to hold the invariant constant.) The circles fitted with this metric to the data of Figure 1 are indistinguishable from the best geometric fits.

One way to visualize the effect of this invariant is to picture an inverted cone (apex at top) with vertical axis and 45-degree side. This cone may be translated arbitrarily (three degrees of freedom) to intersect the plane in an arbitrary circle. The normal paraboloid for that circle is the one which is tangent to the cone at the circle; our invariant finds exactly that paraboloid.

For spheres in R^3, $q(x, y, z) = A(x^2 + y^2 + z^2) + Dx + Ey + Fz + G = 0$, for which the corresponding form to be held constant is $D^2 + E^2 + F^2 - 4AG$, and similarly for higher dimensions.

Our normalization, like Bookstein's (taking his to be $\sqrt{A}$), is a *Euclidean invariant*, that is, its quotient with Euclidean distance is invariant, under similarities – translations, rotations, and scalings – of the plane. Bookstein argues that a normalization containing D, E, or F cannot be invariant even under translations because these quantities can individually grow without bound. The inapplicability of this argument to our normalization should be clear: although D and E can indeed grow without bound, their growth can nevertheless be cancelled by the growth of $4AF$.

Bookstein also argues that a normalization which is not positive-definite will fail to fit data lying exactly on certain curves. This seems to assume that when such a normalization passes through zero on its way to becoming negative it must represent an exact fit to something. The inapplicability of Bookstein's argument to our normalization follows from the fact that $D^2 + E^2 - 4AF$ is nonpositive only for circles of zero radius, which is not the exact or even best fit for any sample not having infinitely many exact fits.

The main adverse effect with our invariant arises from the radial curvature of the paraboloid, i.e. its departure from a cone. For scattered data $|\nabla q|$ will be larger for more outlying data, increasing linearly with distance from the axis of the paraboloid. This makes the fit more responsive to data lying further out, which tends to decrease the curvature of our fit. However this is a second order phenomenon, being tied to the radial curvature of the paraboloid, as opposed to the first order phenomenon illustrated by Figure 1, which depends on the value of $|\nabla q|$ itself rather than its radial variation.

The other drawback of our invariant is that it involves the extraction of eigenvectors. Bookstein's invariant $A = 1$ is simple enough to be used with our Simple Fit algorithm, taking the A coordinate to be the final basis element, which the nonpermuting version of our algorithm automatically sets to 1. The permuting version of our algorithm typically picks some other coefficient to be 1; in the case of the data of Figure 1 it picks E, the coefficient of y, which has the effect of yielding circles whose radius is larger than that of

the best fitting circle by factors of 1, 1.03, 1.1, 1.5 respectively (top to bottom), rather than smaller. However in the absence of a good understanding of what Permuting Simple Fit is up to we remain uncertain as to its reliability for circle fitting.

The conclusion then is, for safety use the invariant $D^2 + E^2 - 4AF$, but if you are willing to take risks then use either Simple Fit with Bookstein's invariant, or Permuting Simple Fit which supplies its own incompletely understood invariant.

Application to Conics and Quadrics. Our method for circles does not generalize directly to ellipses and other conics, since these lack the circular symmetry on which our method depended. This problem is addressed by Sampson's iterative method [24], as we saw at the end of Section 4. In this method Sampson repeatedly fits a conic using Bookstein's invariant $A^2 + B^2/2 + C^2$, at each stage weighting each point p by $1/|\nabla q(p)|$ using the q fitted at the previous stage.

The difficulty we have observed with Bookstein's invariant for circles carries over to conics: conics of low curvature are avoided. Hence this problem can be inherited by Sampson's algorithm. We cure this by showing how to generalize our circle solution to conics.

Our solution is to use the basis $x^2 + y^2, 2xy, x^2 - y^2, x, y, 1$. The advantage of this basis over the customary conic basis $x^2, xy, y^2, x, y, 1$ is that it more cleanly separates out the circular (rotationally invariant) component of a conic, namely the $x^2 + y^2$ basis element. Taking the coefficients to be $A(x^2 + y^2) + 2Bxy + C(x^2 - y^2) + Dx + Ey + F$, we may continue to use our invariant $D^2 + E^2 - 4AF$, simply ignoring the coefficients B and C The effect is as though we had a circle whose diameter is in between the lengths of the major and minor axes, in the case of an ellipse; this is adequate to get a good initial fit of a conic. (Note that if B and C are normalized to $B^2 + C^2 = 1$ then they are respectively $\sin(2\theta)$ and $\cos(2\theta)$ where θ is the orientation of an axis of the conic to the X axis. This is where the 2 in $2xy$ is used.)

The generalization of this basis to higher dimensions is to take the sum of the squares (in four dimensions: $w^2 + x^2 + y^2 + z^2$) along with the differences of consecutive squares ($w^2 - x^2$, $x^2 - y^2$, $y^2 - z^2$) and the cross terms ($2wx$, $2wy$, $2wz$, $2xy$, $2xz$, $2yz$), along with all degree 0 and 1 terms $(w, x, y, z, 1)$.

8. Blend Fit

Given a set $b_i = 0$, $i = 1, \ldots, k$, of *base* surfaces, a *blending surface* is a surface tangent to all of them. (In two dimensions substitute curve for surface.) The problem of finding blending surfaces has received considerable attention in the literature. Some particularly interesting recent approaches are those of Middleditch and Sears [14] and Hoffman and Hopcroft [10,11].

In this section we first describe a new method of constructing blending surfaces that generalizes both the Middleditch-Sears and Hoffman-Hopcroft methods. We then apply this construction to give a method of least-squares fitting of such surfaces; however the construction should prove to be of considerable utility in the theory and applications of blending surfaces independent of our fitting application.

The principle behind our construction can be understood in 2D by considering the lines $Z(x)$ and $Z(y)$, respectively the Y-axis and the X-axis. The zeros of any linear combination $\alpha x + \beta y$ will pass through the intersection of $Z(x)$ and $Z(y)$, but need not be tangent to either of these lines there. However the zeros of any linear combination $\alpha x^2 + \beta y$ with $\beta \neq 0$ will be tangent to $Z(y)$ (consider the curve $y = \alpha x^2$ for any α). The principle is that αx^2 initially grows more slowly with movement away from $Z(x)$ than does βy with movement away from $Z(y)$, provided $\beta \neq 0$. Hence in the neighborhood of the intersection the zeros of $\alpha x^2 + \beta y$ will tend to "stick" to $Z(y)$. The higher the power x^γ the less "sticky" is $Z(x^\gamma)$. This principle generalizes to two arbitrary polynomials in $x_1, \ldots, x_d$ in place of x and y; raising the first to a sufficiently high power will make it negligibly sticky compared to the second at the intersection of their respective zeros, whence the zeros of their linear combination will stick to the second.

For our blending surface construction the two polynomials are the product $\prod_i b_i$ of polynomials defining the base surfaces, and a polynomial t, defining the *truncating surface*, which intersects each base surface in the point(s) of tangency of the blending surface to that base surface. Then by the above principle there is an integer γ large enough that the zeros of any linear combination $\alpha t^\gamma + \beta \prod_i b_i$, $\beta \neq 0$, will be tangent to each b_i where $Z(b_i)$ intersects $Z(t)$.

The canonical example of this in the plane is given by the conic spline, which is a conic section inscribed in a triangle ABC, tangent to AB at A and to BC at C. If a, b, c are linear combinations of $x, y, 1$ such that their respective zeros are the lines BC, CA, and AB, then such conics are given by the zeros of the linear combinations of b^2 and ac. Here $Z(b)$ is the truncating surface, or rather line, and $Z(a), Z(c)$ are the two base lines.

A more interesting example, in 3D, is given by the problem of finding a blending surface (fillet) between equal-diameter cylinders $x^2 + z^2 = 1$ and $y^2 + z^2 = 1$ (unit radius cylinders along the Y and X axes respectively). We take $Z(t)$ to be the ellipsoid $\lambda x^2 + \mu y^2 + z^2 = 1$, where λ and μ are reals less than unity. This ellipsoid is tangent to both cylinders where they intersect the Z-axis, and otherwise intersects each cylinder in the curve where the blending surface will be tangent to that cylinder, with λ and μ providing some variety in the choice of curve. The blending surface is then the degree 4 surface $Z(\alpha(\lambda x^2 + \mu y^2 + z^2 - 1)^2 + \beta(x^2 + z^2 - 1)(y^2 + z^2 - 1))$, with α/β determining how "fat" the fillet is: larger is fatter (more metal if the fillet were a weld).

This generalizes the Middleditch-Sears method by allowing t to be arbitrary; Middleditch and Sears restrict t to be a linear combination of $b_1, \ldots, b_k, 1$ (the 1 being essential), which rules out the truncating surface we used to solve the above cylinder-blending problem. It also considerably simplifies the Hoffman-Hopcroft *potential method* [10], in particular eliminating the complexity in the case when the intersection curve is reducible (e.g. with the above equal-diameter-cylinders problem), as well as generalizing it by permitting more than two surfaces to be blended simultaneously.

Given this notion of a blending surface we turn to the problem of finding such a surface tangent to a given set of base surfaces that best fits a given set of data points. For example we may have two rods welded together, along with a large number of measurements of the

fillet between them, and we want to reduce these samples to a good analytical model of the fillet. This includes discovery of the appropriate truncating surface $Z(t)$; in the rod-blending example we would assume that it was an ellipsoid, leaving only λ and μ to be found in order to determine t, corresponding to selecting a surface having the shape of order 3 generated by the basis $x^2, y^2, z^2 - 1$.

A weaker version of this problem assumes that the truncating surface is completely specified, as for example in [15]. This is not always a good assumption. While it is usually easy to determine the base surfaces – they are typically either given or are large enough as to be easily measured – the exact points of intersection of a sampled blending surface with the base surfaces are not so easily measured, since these points can move a long distance under a very small perturbation of the blending surface. Tangency is an inherently unstable condition in this respect.

We give a very simple method for choosing t of a given shape of order n so as to get the blending surface of best fit. Write the implicit equation of the blending surface as $(\alpha t)^\gamma = \prod_i b_i$. Rewrite it as $\alpha t = (\prod_i b_i)^{1/\gamma}$. Treat this as the problem of fitting the shape whose ideal has the $n+1$ basis functions $t_1, \ldots, t_n, (\prod_i b_i)^{1/\gamma}$, where the t_i's are the basis for the truncating shape.

Previously all our ideals consisted of polynomials. We now have an ideal Q (of the ring of all functions $q : R^d \to R$) containing the nonpolynomial $(\prod_i b_i)^{1/\gamma}$. The beauty of least-squares fitting is that nothing in the theory depends on what functions appear in Q, just so long as Q forms a finite-dimensional vector space, here $n+1$. Of course we need to be able to compute the functions in order to construct the $m \times (n+1)$ matrix A, but $(\prod_i b_i)^{1/\gamma}$ is easily evaluated at each of the m sample points. We also need to be sure that the functions in the basis are independent; it is easily seen that this will be the case if there is only one nonpolynomial in the set and the polynomials form an independent set, which describes the case at hand.

In the case of conics we are given tangents $Z(a)$ and $Z(c)$, a and c being linear combinations of $x, y, 1$, and seek a linear combination b of $x, y, 1$ such that $Z(\alpha b^2 + \beta ac)$ best fits a given set of data points. In this case the above rather dry algebraic solution to this problem has a beautiful geometric visualization. If we take a, v, c to be the coordinates of a 3D space then the 3D surface $v^2 = 4ac$ turns out to be the cone illustrated in Figure 2 (which is taken from Figure 2 of [22], where we give a relatively novel analytic treatment of conic sections by treating them literally as plane sections of a cone).

For each data point p let $a(p)$ denote the value of a at p (the result of evaluating a given the x and y coordinates of p) and similarly for $c(p)$. Each such point then corresponds to a pair of points $(a(p), \pm v, c(p))$ on the cone, obtainable as $v = \pm\sqrt{4a(p)c(p)}$. Discard the $-v$ point. The resulting points, ranging over all the given data, should now approximate a plane in AVC space if as points in XY space they approximate a conic. Then the equation $b = 0$ of this plane yields the desired b. A conic of good fit is obtained by finding the plane of best fit. This is the geometric description of our method for the case of conics.

It will be noted that the method is more sensitive to noise in points in the neighborhood of either tangent. This is due to the cone being steeper (treating V as up) near the tangents.

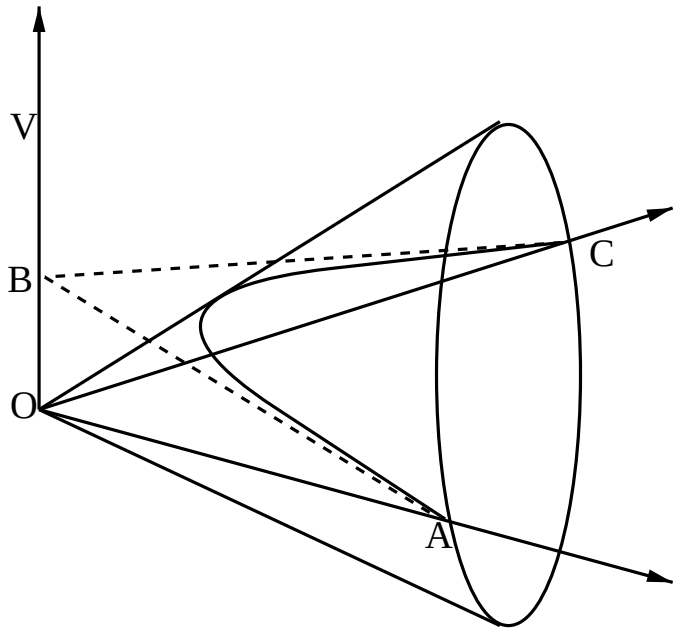

Figure 2. The AVC cone and ABC conic.

This increased sensitivity there corresponds to looking at the points more closely as they approach the tangent in order to tell exactly where the point of tangency is. This insight into the stability of the method is very easily deduced from this geometric picture of our fitting process.

Additional Constraints. Sometimes the truncating surface t will be partially constrained, e.g. one or more points or curves of tangency may be given. When such a constraint can be represented as a linear dependence between the coefficients to be found, the dependence can be used to reduce the order of the shape of the truncating surface, thereby transforming the fitting problem to a simpler one. This is the situation that obtains when either or both of the points of tangency are known when fitting a conic given two of its tangents.

Typography Application. Our application for blend fit has been as part of a two-stage process for reconstructing font outlines from scanned-in images. The first stage finds tangents at lines, extrema, inflexion points, and other suitable articulation points. The second stage then fills in the remainder of the outline by finding the best fitting conic splines as blending curves to these tangents.

Acknowledgments

I am grateful to Gene Golub for some informative discussions.

Bibliography

[1] A. Albano, Representation of Digitized Contours in Terms of Conic Arcs and Straight-Line Segments, *Computer Graphics and Image Processing* **3**, 23-33, 1974.

[2] R.H. Biggerstaff, Three Variations in Dental Arch Form Estimated by a Quadratic Equation, *Journal of Dental Research* **51**, 1509, 1972.

[3] F.L. Bookstein, Fitting Conic Sections to Scattered Data, *Computer Graphics and Image Processing* **9**, 56-71, 1979.

[4] 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 Transactions on Computers* **C-25**, 10, 1020-1032, October 1976.

[5] C. de Boor, **A Practical Guide to Splines**, Springer-Verlag, 1978.

[6] I.D. Faux and M.J. Pratt, **Computational Geometry for Design and Manufacture**, Ellis Horwood, 1978.

[7] Y. Gardan, **Numerical Methods for CAD**, MIT Press, 1986.

[8] A.A. Giordano and F.M. Hsu, **Least Squares Estimation with Applications to Digital Signal Processing**, Wiley, 1985.

[9] R. Gnanadesikan, **Methods for Statistical Data Analysis of Multivariate Observations,** Wiley, 1977.

[10] C. Hoffman and J. Hopcroft, Automatic Surface Generation in Computer Aided Design, *The Visual Computer*, **1**, 2, 92-100 (1985).

[11] C. Hoffman and J. Hopcroft, Quadratic Blending Surfaces, *Computer Aided Design*, **18**, 6, 301-306 (Jul-Aug. 1986).

[12] C.L. Lawson and R.J. Hanson, **Solving Least-Squares Problems**, Prentice-Hall, 1974.

[13] E.A. Lord and C.B. Wilson, **The Mathematical Description of Shape and Form**, Ellis Horwood, Chichester, 1984.

[14] A.E. Middleditch and K.H. Sears, Blend Surfaces for Set Theoretic Volume Modelling Systems, *Computer Graphics* **19**, 3 (Siggraph-85), 161-170, July 1985.

[15] V.S. Nalwa and E. Pauchon, Edgel-Aggregation and Edge-Description, *Eighth International Conference on Pattern Recognition,* 604-609, Paris, Oct. 1986,

[16] K. Paton, Conic Sections in Chromosome Analysis, *Pattern Recognition* **2**, 39-51, 1970.

[17] T. Pavlidis, Curve Fitting with Conic Splines, *ACM Transactions on Graphics* **2**, 1, 1-31, January 1983.

[18] K. Pearson, On lines and planes of closest fit to systems of points in space, *Philos. Mag. Ser. 6*, 2, 559, 1901.

[19] C. Pearson, **Numerical Methods in Engineering and Science**, Van Nostrand Reinhold, 1986.

[20] M.A. Penna and R.R. Patterson, **Projective Geometry and its Applications to Computer Graphics**, Prentice-Hall, New Jersey, 1986.

[21] M. Plass and M. Stone, Curve-Fitting with Piecewise Parametric Cubics, *Computer Graphics* **17**, 3 (Siggraph-83), 229-239, July 1983.

[22] V. Pratt, Techniques for Conic Splines, *Computer Graphics* **19**, 3 (Siggraph85), 151-159, July 1985.

[23] D. Proffitt, The Measurement of Circularity and Ellipticity on a Digital Grid, *Pattern Recognition* **15**, 5, 383-387, 1982.

[24] P.D. Sampson, Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm, *Computer Graphics and Image Processing* **18**, 97-108, 1982.

[25] K. Turner, Computer Perception of Curved Objects using a Television Camera, Ph.D. Thesis, Dept. of Machine Intelligence, University of Edinburgh, Nov. 1974.