Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation

Gabriel Taubin, *Member, IEEE*

Abstract— **This paper addresses the problem of parametric representation and estimation of complex planar curves in 2-D, surfaces in 3-D and nonplanar space curves in 3-D. Curves and surfaces can be defined either parametrically or implicitly, and we use the latter representation. A planar curve is the set of zeros of a smooth function of two variables x-y, a surface is the set of zeros of a smooth function of three variables x-y-z, and a space curve is the intersection of two surfaces, which are the set of zeros of two linearly independent smooth functions of three variables x-y-z. For example, the surface of a complex object in 3-D can be represented as a subset of a single implicit surface, with similar results for planar and space curves. We show how this unified representation can be used for object recognition, object position estimation, and segmentation of objects into meaningful subobjects, that is, the detection of "interest regions" that are more complex than high curvature regions and, hence, more useful as features for object recognition. Fitting implicit curves and surfaces to data would be ideally based on minimizing the mean square distance from the data points to the curve or surface. Since the distance from a point to a curve or surface cannot be computed exactly by direct methods, the approximate distance, which is a first-order approximation of the real distance, is introduced, generalizing and unifying previous results. We fit implicit curves and surfaces to data minimizing the approximate mean square distance, which is a nonlinear least squares problem. We show that in certain cases, this problem reduces to the generalized eigenvector fit, which is the minimization of the sum of squares of the values of the functions that define the curves or surfaces under a quadratic constraint function of the data. This fit is computationally reasonable to compute, is readily parallelizable, and, hence, is easily computed in real time. In general, the generalized eigenvector fit provides a very good initial estimate for the iterative minimization of the approximate mean square distance. Although we are primarily interested in the 2-D and 3-D cases, the methods developed herein are dimension independent. We show that in the case of algebraic curves and surfaces, i.e., those defined by sets of zeros of polynomials, the minimizers of the approximate mean square distance and the generalized eigenvector fit are invariant with respect to similarity transformations. Thus, the generalized eigenvector fit is independent of the choice of coordinate system, which is a very desirable property for object recognition, position estimation, and the stereo matching problem. Finally, as applications of the previous techniques, we illustrate the concept of "interest regions" for object recognition and describe a variable-order segmentation algorithm that applies to the three cases of interest.**

Index Terms—**Algebraic curves and surfaces, approximate distances, generalized eigenvector fit, implicit curve and surface fitting, invariance, object recognition, segmentation.**

Manuscript received April 13, 1988; revised April 30, 1991. This work was supported by an IBM Fellowship and NSF Grant IRI-8715774.

The author was with the Laboratory for Engineering Man/Machine Systems, Department of Engineering, Brown University, Providence, RI 02912. He is now with the Exploratory Computer Vision Group, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 9102093.

I. Introduction

OBJECT recognition and position estimation are two central issues in computer vision. The selection of an internal representation for the objects with which the vision system has to deal, and good adaptation for these two objectives, is among the most important hurdles to a good solution. Besl [5] surveys the current methods used in this field. The early work on recognition from 3-D data focused on polyhedral objects. Lately, quadric patches have been considered as well [34], [33], [9], [10]. Bolle and Vemuri [11] survey the current 3-D surface reconstruction methods. Among the first systems using planar curves extracted from edge information in range data is 3DPO [13], [12]. Nonplanar curves arise naturally when curved surfaces are included in the models and have been represented as parameterized curves in the past [65], that is, $x(t), y(t), z(t)$ are explicit functions of a common parameter t.

At least the following three reasons, illustrated in Fig. 1, justify our interest in representing and estimating nonplanar curves:

1) The intersection of two 3–D surface patches\newline is usually a nonplanar curve.
2) If the object has patterns on the surfaces, the boundaries of a pattern are usually nonplanar curves.
3) For objects consisting of many small smooth surface patches, estimating nonplanar curves of surface intersections may be more useful than estimating the surface patches.

There is no reason to restrict the surfaces considered to be quadric surfaces. In general, we will represent a surface as the set of roots of a smooth implicit function of three variables x-y-z, a space curve as the intersection of two different surfaces, and a planar curve as the set of roots of a smooth function of two variables x-y. In this way, a 3-D object will be represented either as a set of surface patches, as a set of surface patches specifying curve patches, or as a combination of both. A 2-D

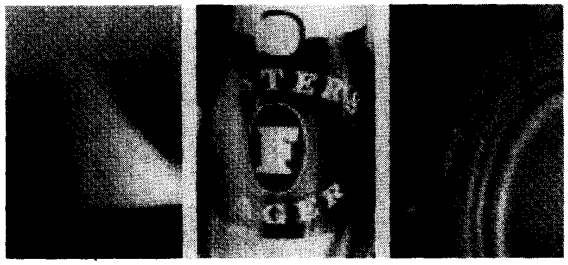


Fig. 1. Reasons for estimating nonplanar curves.

object will be represented as a set of planar curve patches.

The representation of curves and surfaces in implicit form has many advantages. In the first place, an implicit curve or surface maintains its implicit form after a change of coordinates, that is, if a set of points can be represented as a subset of an implicit curve or surface in one coordinate system so can it be in any other coordinate system. That is not the case with data sets represented as graphs of functions of two variables. In the second place, the union of two or more implicit curves or surfaces can be represented with a single implicit curve or surface. This property is very important in relation with the segmentation problem and the concept of "interest regions," which are regions more complex than high curvature regions that can be used as features for recognition.

Given a family of implicit functions parameterized by a finite number of parameters and a finite set of points in space assumed to belong to the same surface or curve, we want to estimate the parameters that minimize the mean square distance from the data points to the surface or curve defined by those parameters. Unfortunately, there is no closed-form expression for the mean square distance from a data set to a generic curve or surface, and iterative methods are required to compute it.

In this paper, we develop a first-order approximation for the distance from a point to a curve or surface generalizing some previous results. The mean value of this function on a fixed set of data points is a nonlinear function of the coefficients, but since it is a smooth function of these coefficients, it can be minimized using well-established nonlinear least squares techniques. However, since we are interested in the global minimum and these numerical techniques find local minima, a good initial estimate is required.

In the past, other researchers have minimized a different mean square error: the mean sum of squares of the values of the functions that define the curve or surface on the data points under different constraints. It is well known that this performance function can produce a very biased result. We study the geometric conditions under which the curve or surface produced by the minimization of the mean square error fails to approximate the minimizer of the approximate mean square distance. This analysis leads us to a quadratic constraint (a function of the data) that turns the minimization of the mean square error into a stable and robust generalized eigenvector problem in the linear case, that is, when the admissible functions form a vector space. For example, algebraic curves and surfaces of arbitrary degree can be fitted with this method.

We then introduce the reweight procedure, which in most of the cases helps to improve the solution produced by the generalized eigenvector fit at a lower cost than the general iterative minimization techniques. Finally, the result of the reweight procedure is fed into the Levenberg-Marquardt algorithm in order to minimize the approximate mean square distance.

In the case of algebraic curves and surfaces, the results of these minimization processes enjoy the very desirable property of being invariant with respect to similarity transformations of the data set, particularly with respect to rigid body transformations. Hence, these fits are independent of the coordinate system used.

In Section II, we define implicit curves and surfaces and explain some of their properties. In Section III, we show that complex objects can be represented as subsets of a single curve or surface and that this representation unifies the problems of image segmentation, position estimation, and object recognition. In Section IV, we derive the approximate distance, which is the first-order approximation to the real distance, from a point to a curve or surface. In Section V, we introduce the approximate square distance and develop the constraints for the linear case. In Section VI, we study the relation between the mean square error and the approximate square distance, establishing the relation of our contribution with the previous work. In Section VII, we introduce the generalized eigenvector fit method for the linear case, and in Appendix B, we analyze the existence and uniqueness of the solution; in Section VIII, we analyze its complexity. In Section IX, we show that the curves and surfaces produced by the generalized eigenvector fit and the minimization of the approximate mean square distance are invariant under change of basis in the linear case and under similarity transformations in the case of algebraic curves and surfaces. In Section X, we introduce the reweight procedure and establish its relation with previous work. In Section XI, we describe several families of parameterized implicit curves and surfaces, including superquadrics, where the methods introduced in this paper can be applied. In Section XII, we survey the previous work on implicit curve and surface fitting, establishing their relation with the methods introduced in this paper. In Section XIII, we introduce the concept of "interest region," and we briefly explain how it could be used for object recognition in a cluttered environment. Finally, in Section XIV, we describe a variable-order algorithm for the segmentation of curves and surfaces in terms of algebraic primitives, and in Section XV, we describe the experimental results.

II. Implicit Curves and Surfaces

Let $f : \mathbb{R}^n \to \mathbb{R}^k$ be a *smooth* map, a map with continuous first- and second-order derivatives at every point. We say that the set $Z(f) = \{x : f(x) = 0\}$ of zeros of f is defined by the *implicit equations*

$$f_1(x) = 0, \ldots, f_k(x) = 0.$$

We are interested in three particular cases for their applications in computer vision and computer-aided design. They have

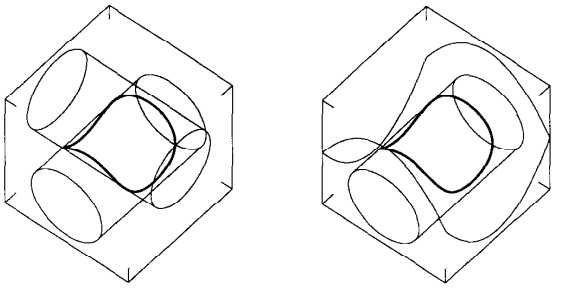


Fig. 2. Different representations of the same space curve.

special names: $Z(f)$ is a *planar curve* if $n = 2$ and $k = 1$, it is a *surface* if $n = 3$ and $k = 1$, and it is a *space curve* if $n = 3$ and $k = 2$. In order to avoid pathological cases, we have to require that the set of points of $Z(f)$ that are *regular points* of f be dense in $Z(f)$, where a point $x \in \mathbb{R}^n$ is a *regular* point of f if the Jacobian matrix

$$Df(x) = \begin{pmatrix} \frac{\delta f_1}{\delta x_1}(x) & \cdots & \frac{\delta f_1}{\delta x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\delta f_k}{\delta x_1}(x) & \cdots & \frac{\delta f_k}{\delta x_n}(x) \end{pmatrix} = \begin{pmatrix} \nabla f_1(x)^t \\ \vdots \\ \nabla f_k(x)^t \end{pmatrix}$$

has rank k or, equivalently, if the matrix $Df(x)Df(x)^t$ is nonsingular. Otherwise, x is a *singular* point of f.

The intersection of two surfaces is a space curve. However, this representation is not unique; a space curve can be represented as the intersection of many pairs of surfaces. For example, if $Z(f)$ is the intersection of two cylinders

$$f(x) = \begin{pmatrix} x_1^2 + (x^3 - 1)^2 - 4 \\ x_2^2 + (x^3 + 1)^2 - 4 \end{pmatrix} = \begin{pmatrix} -3 - 2x_3 + x_3^2 + x_1^2 \\ -3 - 2x_3 + x_3^2 + x_2^2 \end{pmatrix}$$

and

$$g(x) = \begin{pmatrix} -3 - 2x_3 + x_3^2 + x_1^2 \\ 4x_3 + x_2^2 - x_1^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} f(x)$$

the second component of g represents a hyperbolic paraboloid, and the sets of zeros of f and g are exactly the same: $Z(f) = Z(g)$. Fig. 2 shows these two different representations of the same space curve. In general, for any *nonsingular* $k \times k$ matrix A, the function $g = Af$ has the same zeros as f does: $Z(Af) = Z(f)$. If $g = Af$ for certain nonsingular $k \times k$ matrix A, we say that f and g are two *representations* of the same set $Z(f)$. Particularly, for $k = 1$, the planar curve or surface $Z(f)$ is identical to $Z(\lambda f)$ for every nonzero λ.

Unions of implicit surfaces are implicit surfaces. For example, the union of the two cylinders of the previous example

$$\{x : x_1^2 + (x_3 - 1)^2 - 4 = 0\} \cup \{x : x_2^2 + (x_3 + 1)^2 - 4 = 0\}$$

is the surface defined by the set of zeros of the product

$$\{x : (x_1^2 + (x_3 - 1)^2 - 4)(x_2^2 + (x_3 + 1)^2 - 4) = 0\}.$$

Hence, a single fourth-degree polynomial can represent a pair of cylinders, and this is true for arbitrary cylinders, e.g., a pair that do not intersect. Note that although the two cylinders are regular surfaces, the points that belong to the intersection curve become singular points of the union, the normal vector to the surface is not uniquely defined on the curve. In general, if $Z(f_1), \ldots, Z(f_k)$ are surfaces, their union is the set of zeros of the product of the functions

$$Z(f_1) \cup \cdots \cup Z(f_k) = Z(f_1 \cdots f_k)$$

with the points that belong to the intersection curve

$$\bigcup_{i \neq j} Z(f_i) \cap Z(f_j)$$

being singular points of the union. The same results hold for planar curves. The case of space curves is more complicated, but, for example, the union of two implicit space curves $Z(f) \cup Z(g)$ is *included* in the space curve

$$\{x : f_1(x)g_1(x) = 0, f_2(x)g_2(x) = 0\}.$$

III. Object Representation, Segmentation and Recognition

The boundaries of most manufactured objects can be represented exactly as piecewise smooth surfaces, which in turn can be defined by implicit equations, usually algebraic surfaces.

Given a parameterized family of implicit surfaces, *image segmentation* is the problem of finding a partition of a data set into homogeneous regions, where each of them are well approximated by a member of the family under certain approximation criteria.

An object $\mathcal{O}$ is a collection of surface patches

$$\mathcal{O} = \bigcup_{i=1}^{q} \mathcal{O}_i,$$

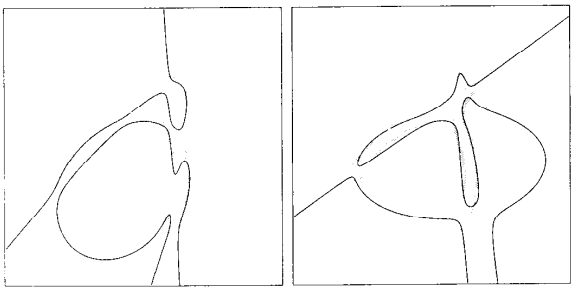
where each surface patch is a regular *subset* of an implicit surface

$$\mathcal{O}_i \subseteq Z(f_i) = \{x : f_i(x) = 0\} \qquad i = 1, \ldots, q.$$

A subcollection of patches, or even the whole object, can be represented as a subset of a *single implicit surface*

$$\mathcal{O} \subseteq Z(f_1 \cdots f_q) = \bigcup_{i=1}^{q} Z(f_i).$$

The boundaries of many more objects can be *approximated* with piecewise algebraic surfaces. An object $\mathcal{O}$ is represented approximately as a subset of a single implicit surface $Z(f)$ under certain approximation criterion. The 2-D parallel of this representation allows us to represent sets of picture edges as subsets of a single implicit planar curve, and the set of space curves corresponding to surface normal discontinuities, or other geometric invariant curves such as the the lines of curvature of surfaces [74], [73] can be approximated by a subset of a single implicit space curve. This single implicit curve or surface will be called a *model* of $\mathcal{O}$ in *standard position.* Fig. 3 shows an attempt to recover the boundary



(a) (b)

Fig. 3. Degree 5 planar curve fit: (a) Generalized eigenvector fit; (b) after reweight procedure.

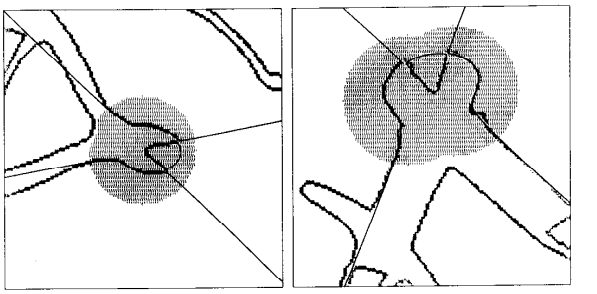


Fig. 4. Regions well approximated by fourth-degree algebraic curves. Only the data inside the gray areas have been used in the computations; therefore, the approximation is good only there.

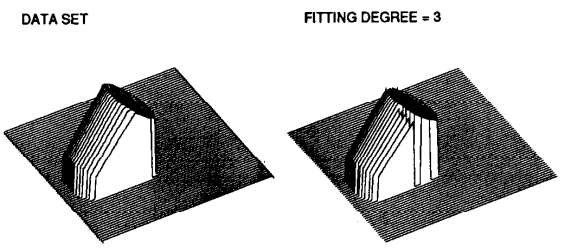


Fig. 5. Original range data and a single third-degree polynomial fit to the planar cap and patch of cylindrical side of a pencil sharpener.

of an object with some of the methods to be introduced in subsequent sections. The set of zeros of a single fifth-degree polynomial is fitted to the data using the generalized eigenvector fit algorithm of Section VII, and then the fit is improved with the reweight procedure of Section X. Although the curve does not fit the data well because the degree is not high enough, we can appreciate how the fit is refined by the iterative reweight procedure. Since from a practical point of view, it is not convenient to work with very-high-degree polynomials, we see this unified representation as a way to represent small groups of smooth patches. Based on this idea, in Section XIII, we introduce the concept of *interest region*, and we sketch a tentative approach to object recognition in a cluttered environment. For example, in Fig. 4, we show how the tip of the pliers shown in Fig. 3 can be well approximated by a fourth-degree algebraic curve, and in Fig. 5, we show the result of fitting a single third-degree algebraic surface to two visible surface patches of a pencil sharpener, a planar end, and a patch of cylindrical side.

If the data set is an observation of a part of a single and known object $\mathcal{O}$ in an unknown position and $Z(f)$ is a model of $\mathcal{O}$ in standard position, then *position estimation* becomes the problem of best fitting the data set with a member of the family of implicit curves or surfaces defined as compositions of f with elements of the admissible family of transformations $\mathcal{G}$ because if $T \in \mathcal{G}$ is a nonsingular transformation, then

$$\begin{aligned} T^{-1}[Z(f)] &= \{T^{-1}(y) : f(y) = 0\} \\ &= \{x : f(T(x)) = 0\} \quad = Z(f \circ T) \end{aligned}$$

that is, the implicit curve or surface $Z(f)$ transformed by T^{-1} is a new implicit curve or surface, where the curve or surface is defined as the set of zeros of f composed with the transformation T.

Typical examples of families of transformations are rigid body, similarity, affine, and projective transformations. A detailed formulation is given in Section XI. If the object is unknown but it is known that it is one of a finite number of known objects modeled in standard position by the curves or surfaces $Z(f_1), \ldots, Z(f_q)$, then *object recognition* is equivalent to estimating the position of each object, assuming that the data corresponds to that object and then associating the data to the object that minimizes the fitting criterion.

If the data set is a view of several objects, object recognition is equivalent to segmentation with curve or surface primitives belonging to the family of compositions of models of known objects in standard position with admissible transformations.

Two different approximation criteria will be considered in different parts of this paper. They are based on the 2-norm

$$\frac{1}{q}\sum_{i=1}^{q} \operatorname{dist}(p_i, Zf)^2$$

and the ∞-norm

$$\sup_{1 \le i \le q} \operatorname{dist}(p_i, Z(f))$$

where $\mathcal{D} = \{p_1, \ldots, p_q\}$ is a finite data set, and $\operatorname{dist}(p_i, Z(f))$ is the distance from the point p_i to the curve or surface $Z(f)$.

We are primarily interested in fitting curves and surfaces to data under the ∞-norm, but fitting under the 2-norm is computationally less expensive. In Section XIV, we describe an algorithm that follows the classical *hypothesize and test* approach, a curve or surface is *hypothesized* by minimizing an approximation to the 2-norm, and then it is *tested* with the ∞-norm.

IV. Approximate Distance

Since a general formulation lets us study the three cases of interest at once, we will continue our analysis in this way, showing at the same time that it applies to an arbitrary dimension.

In general, the distance from a regular point $x \in \mathbb{R}^n$ of a smooth map $f : \mathbb{R}^n \to \mathbb{R}^k$ to the set of zeros $Z(f)$ cannot

be computed by direct methods. The case of a linear map is an exception, in which case the Jacobian matrix $\mathbf{D} = Df(x)$ is constant, and we have the identity

$$f(y) \equiv f(x) + \mathbf{D}(y - x).$$

Without loss of generality, we will assume that the rank of $\mathbf{D}$ is k. The unique point $\hat{y}$ that minimizes the distance $\|y - x\|$ to x, constrained by $f(y) = 0$, is given by

$$\hat{y} = x - \mathbf{D}^{\dagger} f(x)$$

where $\mathbf{D}^{\dagger}$ is the *pseudoinverse* (see section 9.2 of [29] and chapter 6 of [42]) of $\mathbf{D}$. In our case, $\mathbf{D}^{\dagger} = \mathbf{D}^t(\mathbf{D}\mathbf{D}^t)^{-1}$, with the square of the distance from x to $Z(f)$ being

$$\operatorname{dist}(x, Zf)^2 = \|\hat{y} - x\|^2 = f(x)^t\left(\mathbf{D}\mathbf{D}^t\right)^{-1} f(x).$$

In the nonlinear case, we approximate the distance from x to $Z(f)$ with the distance from x to the set of zeros of a *linear model* of f at x, which is a linear map $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^k$ such that

$$f(y) - \tilde{f}(y) = O(\|y - x\|^2).$$

Such a map is uniquely defined when x is a regular point of f, and it is given by the truncated Taylor series expansion of f

$$\tilde{f}(y) = f(x) + Df(x)(y - x).$$

Clearly $\tilde{f}(x) = f(x), D\tilde{f}(x) = Df(x)$, and we have

$$\operatorname{dist}(x, Zf)^2 \approx f(x)^t\left(Df(x)Df(x)^t\right)^{-1} f(x). \qquad (1)$$

This normalization generalizes two previous results. For $k = 1$, which is the case of planar curves and surfaces, the Jacobian has only one row $Df(x) = \nabla f(x)^t$, and the right-hand side member of (1) reduces to $f(x)^2/\|\nabla f(x)\|^2$, where the value of the function is scaled down by the rate of growth at the point. Turner [72] and Sampson [64] have independently proposed it for particular cases of curve fitting. For $k = n$, the Jacobian matrix is square and nonsingular, and therefore, the right-hand side member of (1) reduces to $\|Df(x)^{-1} f(x)\|^2$, which is the length of the update of the Newton-Raphson root finding algorithm (see chapter 5 of (27)]

$$x' = x - Df(x)^{-1} f(x).$$

Our contribution is the extension to space curves and, in general, to curves and surfaces of any dimension.

Since the right-hand side member of (1) is a nonnegative number, from now on, we will call

$$\sqrt{f(x)^t(Df(x)Df(x)^t)^{-1} f(x)}$$

the *approximate distance* from x to $Z(f)$. Fig. 6 shows several contours of constant distance, constant function value, and constant approximate distance for the simplest case of a planar curve with a singular point: the pair of intersecting lines $\{x : x_1 x_2 = 0\}$. Fig. 7 shows the same contours for the regular curve $\{x : 8x_1^2 + (x_2^2 - 4)^2 - 32 = 0\}$. The contours of constant function value tend to be farther from the singular

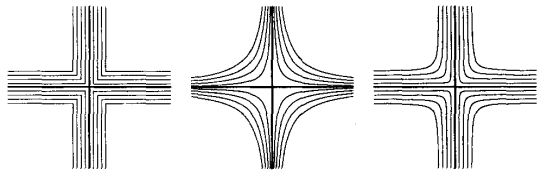


Fig. 6. Contours of constant distance, constant function value, and constant approximate distance to the curve $\{x : x_1 x_2 = 0\}$ near a singular point.

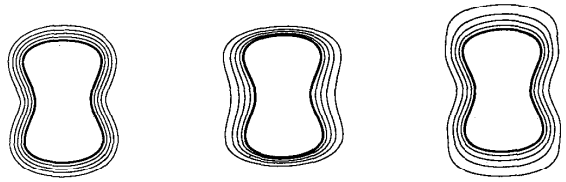


Fig. 7. Contours of constant distance, constant function value, and constant approximate distance to the curve$\{x : 8x_1^2 + (x_2^2 - 4)^2 - 32 = 0\}$ near a regular curve.

points and closer to the regular points than the real distance. The approximate distance solves these problems.

The approximate distance has several interesting geometric properties. It is independent of the representation of $Z(f)$. If A is a nonsingular $k \times k$ matrix and $g(x) = Af(x)$, then

$$\begin{aligned} &g(x)^t\left(Dg(x)Dg(x)^t\right)^{-1} g(x) \\ &\quad = f(x)^t A^t\left(ADf(x)Df(x)^t A^t\right)^{-1} Af(x) \\ &\quad = f(x)^t\left(Df(x)Df(x)^t\right)^{-1} f(x). \end{aligned} \qquad (2)$$

It is also invariant to rigid body transformations of the space variables; if $T(x) = Ux + b$ is a rigid body transformation, then $D\big(f(Ux + b)\big) = Df(Ux + b)U$, and therefore

$$\begin{aligned} &D\big(f(Ux + b)\big)D\big(f(Ux + b)\big)^t \\ &\quad = Df(Ux + b)UU^t Df(Ux + b)^t \\ &\quad = Df(Ux + b)Df(Ux + b)^t. \end{aligned}$$

A similar derivation shows that a scale transformation of the space variables produces the corresponding scaling of the approximate distance.

Since we are interested in fitting curves and surfaces to data in a finite number of steps, we will restrict ourselves to families of maps described by a finite number of parameters. Let us fix a smooth function $\phi : \mathbb{R}^{r+n} \to \mathbb{R}^k$ defined almost everywhere. From now on, we will only consider maps $f : \mathbb{R}^n \to \mathbb{R}^k$, which can be written as

$$f(x) \equiv \phi(\alpha, x)$$

for certain $\alpha = (\alpha_1, \ldots, \alpha_r)^t$, in which case, we will also write $f = \phi_\alpha$. We will refer to $\alpha_1, \ldots, \alpha_r$ as the *parameters* and to $x_1, \ldots, x_n$ as the *variables*. The family of all such maps will be denoted

$$\mathcal{F} = \{f : \exists \alpha f = \phi_\alpha\}.$$

We will say that ϕ is the *parameterization* of the family $\mathcal{F}$, and we will write $\mathcal{F}_\phi$ when having to differentiate among different

parameterizations. The approximate distance from x to $Z(\phi_\alpha)$ will be denoted

$$\delta(\alpha, x) = \sqrt{\phi_\alpha(x)^t (D\phi_\alpha(x) D\phi_\alpha(x)^t)^{-1} \phi_\alpha(x)}$$

or $\delta_\phi(\alpha, x)$ when it is necessary.

We will give special attention to the *linear parameterization*, where $\phi(\alpha, x)$ is a linear function of α, and therefore, $\mathcal{F}$ is a finite dimensional vector space of smooth maps $\mathbb{R}^n \to \mathbb{R}^k$. We will refer to this case as the *linear case*. In the linear case, a map f belongs to $\mathcal{F}$ if and only if for every nonsingular $k \times k$ matrix A, Af also belongs to $\mathcal{F}$. Vector spaces of polynomials of degree $\leq d$ are typical examples of linearly parameterized families. In Section XI, we show several nonlinear parameterizations of families of curves and surfaces with applications in computer vision.

V. Approximate Mean Square Distance

Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be a set of n-dimensional data points, and let $Z(f)$ be the set of zeros of $f = \phi_\alpha : \mathbb{R}^n \to \mathbb{R}^k$. If we assume that α is known and $\delta(\alpha, p_1)^2, \ldots, \delta(\alpha, p_q)^2$ are independent and uniformly distributed as the square of a normal random variable of mean zero and variance σ^2, the sum

$$\frac{1}{\sigma^2} \sum_{i=1}^{q} \delta(\alpha, p_i)^2 \tag{3}$$

has a χ^2 distribution with q degrees of freedom.

If the true α is unknown, it can be estimated minimizing (3). Curve or surface fitting corresponds to the minimization of (3) with respect to the unknown parameters of $\alpha_1, \ldots, \alpha_r$.

Assuming that the variance σ^2 is known, the problem is equivalent to minimizing the *approximate mean square distance* from the data set $\mathcal{D}$ to the set of zeros of $f = \phi_\alpha$

$$\Delta^2_{\mathcal{D}}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} \delta(\alpha, p_i)^2. \tag{4}$$

Depending on the particular parameterization ϕ, the r parameters $\alpha_1, \ldots, \alpha_r$ might not be independent. For example, by (2), the sum does not change if we replace Af for f, where A is a nonsingular $k \times k$ matrix, and if $\hat{f}$ minimizes the approximate mean square distance (4), so does $A\hat{f}$. In other words, the parameters might not be *identifiable*. This lack of identifiability is not a problem, though, because we are not interested in the function $\hat{f}$ but in its set of zeros $Z(\hat{f})$. For example, in the linear case, the symmetric matrix constraint

$$\frac{1}{q} \sum_{i=1}^{q} Df(p_i) Df(p_i)^t = I_k \tag{5}$$

which is equivalent to $k(k+1)/2$ scalar constraints on the parameters, can be imposed on f without affecting the set of zeros of a minimizer of (4). The reason is that $\Delta^2_{\mathcal{D}}(\alpha)$ is defined only if the symmetric matrices

$$Df(p_1) Df(p_1)^t, \ldots, Df(p_q) Df(p_q)^t$$

are nonsingular, and since all of these matrices are also nonnegative definite, they are positive definite, and their mean

$$\frac{1}{q} \sum_{i=1}^{q} Df(p_i) Df(p_i)^t \tag{6}$$

is positive definite as well, in which case there exists a nonsingular $k \times k$ matrix A such that

$$I_k = A\left(\frac{1}{q} \sum_{i=1}^{q} Df(p_i) Df(p_i)^t\right) A^t$$

$$= \frac{1}{q} \sum_{i=1}^{q} D[Af](p_i) D[Af](p_i)^t,$$

and Af satisfies the constraint (5). We can take A to be the inverse of the Cholesky decomposition of (6). Since f belongs to $\mathcal{F}$ if and only if Af does, the constraint (5) does not impose any restriction on the set of admissible curves or surfaces $\{Z(f) : f \in \mathcal{F}\}$.

The problem of computing a *local minimum* of an expression like (4) is known as the nonlinear least squares problem, and it can be solved using several iterative methods (see chapter 10 of [27]). Among them, the Levenberg-Marquardt algorithm [49], [52] is probably the best known, and excellent implementations of it are available in subroutine packages such as MINPACK [53]. A short description of the Levenberg-Marquardt algorithm in the context of our problem is given in Appendix A.

Every local minimization algorithm requires a good starting point. Since we are interested in the *global minimization* of (4), even using the Levenberg-Marquardt algorithm, we need a method to chose a good initial estimate.

VI. Mean Square Error

The study of $\Delta^2_{\mathcal{D}}(\alpha)$ in a particular case will provide us with a good strategy to choose an initial estimate in certain cases, such as, for example, in the linear case. Let us assume that the matrix function $Df(x)Df(x)^t$ is *constant* on the set $Z(f)$; in particular, $Z(f)$ does not have singular points. For $k = 1$, this means that the length of the gradient of the unique component f_1 of f is constant on $Z(f)$, but nothing is said about its orientation. Linear functions obviously have this property because in this case, $Df(x)$ is already constant, but circles, spheres, and cylinders, among other families of functions, have the same property. If f also satisfies the constraint (5) and the data points are close to the set of zeros of f, by continuity of Df, we have

$$I_k = \frac{1}{q} \sum_{i=1}^{q} Df(p_i) Df(p_i)^t \approx Df(p_j) Df(p_j)^t$$

for $j = 1, \ldots, q$, and the approximate mean square distance $\Delta^2_{\mathcal{D}}(\alpha)$ is approximated by the *mean square error*

$$\xi^2_{\mathcal{D}}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} \|f(p_i)\|^2. \tag{7}$$

In this particular case, when $DfDf^t$ is constant on $Z(f)$, the minimizers of (7) and (4), both constrained by (5), are almost the same, and we will see in the following section that in the

linear case, the *global* minimizer of (5)–(7) can be computed at a much lower cost than a local minimizer of (4)–(5).

Furthermore, we have observed that if $\hat{f}$ is the global minimizer of $\Delta^2_{\mathcal{D}}(\alpha)$ and the matrix $D\hat{f}D\hat{f}^t$ is not close to a singular matrix on $\mathcal{D}$, the minimizer of $\xi^2_{\mathcal{D}}(\alpha)$ constrained by (5) is a very good approximation of $\hat{f}$, and the Levenberg-Marquardt algorithm started from this estimate converges quickly after a few iterations. Geometrically, $D\hat{f}D\hat{f}^t$ not close to a singular matrix means that no point of $\mathcal{D}$ is close to a singular point of $\hat{f}$. Since $\hat{f}$ is unknown, we cannot test beforehand whether $D\hat{f}D\hat{f}^t$ is close to a singular matrix or not. In order to speed up the convergence of the Levenberg-Marquardt algorithm, after computing the minimizer of the mean square error, and before the iterative minimization, we apply the *reweight procedure*, which is described after the analysis of the linear case in Section X.

Most of the previous work on fitting implicit curves and surfaces to data has been based on minimizing the mean square error but with different constraints. In Section XII, we give a detailed description of these earlier methods.

VII. Generalized Eigenvector Fit

In this section, we show that in the linear model, the minimization of the mean square error (7) constrained by (5) reduces to a generalized eigenvector problem. In Section XII, we show that this method, introduced by the author [70], [71], generalizes several earlier eigenvector fit methods.

Let $X_1(x), \ldots, X_h(x)$ be linearly independent smooth functions, e.g., polynomials, and let us denote

$$X = (X_1, \ldots, X_h)^t : \mathbb{R}^n \to \mathbb{R}^h.$$

In this section, all the maps can be written as linear combinations of the components of X

$$f = FX : \mathbb{R}^n \to \mathbb{R}^k$$

for a $k \times h$ matrix F of real numbers. The parameter vector α has $r = hk$ elements, and it is equal to the concatenation of the rows of F

$$F_{ij} = \alpha_{(i-1)h+j} \qquad i = 1, \ldots, k \quad j = 1, \ldots, h.$$

Since differentiation is a linear operation, we have

$$Df = D[FX] = F[DX]$$

where DX is the Jacobian of X. The constraint (5) become a quadratic constraint on the elements of F

$$I_k = \frac{1}{q}\sum_{i=1}^{q} F[DX(p_i)][DX(p_i)^t]F^t = FN_{\mathcal{D}}F^t \qquad (8)$$

where

$$N_{\mathcal{D}} = \frac{1}{q}\sum_{i=1}^{q}[DX(p_i)DX(p_i)^t]$$

is symmetric nonnegative definite. The approximate mean square distance $\Delta^2_{\mathcal{D}}(\alpha)$ does not have any special form, but the mean square error (7) becomes

$$\begin{aligned}\xi^2_{\mathcal{D}}(\alpha) &= \frac{1}{q}\sum_{i=1}^{q}\|FX(p_i)\|^2 \\ &= \frac{1}{q}\sum_{i=1}^{q}\operatorname{trace}\bigl(F[X(p_i)X(p_i)^t]F^t\bigr) \\ &= \frac{1}{q}\operatorname{trace}\bigl(FM_{\mathcal{D}}F^t\bigr) \qquad (9)\end{aligned}$$

where

$$M_{\mathcal{D}} = \frac{1}{q}\sum_{i=1}^{q}[X(p_i)X(p_i)^t]$$

which is the covariance matrix of X over the data set $\mathcal{D}$. This matrix is classically associated with the normal equations of the least square method (see chapter 6 of [42]). In addition, several researchers have introduced linear or quadratic constraints on F to fit implicit curves or surfaces $k = 1$ to data [1], [8], [14], [20], [22], [23], [40], [55], [54], [60], [64] minimizing (9). However, all of these constraints do not take into account the data; they are fixed and predetermined constraints on the coefficients, and most of them introduce singularities in parameter space, i.e., certain parameters are *never* solutions of the corresponding method. A detailed description of these methods is given in Section XII. Our contribution is the introduction of the quadratic constraint (8), which is function of the data, and the handling of space curves within the same framework. The generalized eigenvector fit algorithm is more robust than most of the previous direct methods, except perhaps for Pratt's *simple fit* algorithm [60], which is explained in Section XII, which seems to be equivalent both in computational cost and robustness. We plan to carry out a detailed comparison of the generalized eigenvector fit and the simple fit algorithms in the near future.

Note that if $M_{\mathcal{D}}$ is singular and a row F_j of the matrix F belongs to the null space of $M_{\mathcal{D}}$, then

$$0 = F_j M_{\mathcal{D}} F_j^t = \frac{1}{q}\sum_{i=1}^{q}|F_j X(p_i)|^2$$

and the function $f_j = F_j X$ is identically zero on $\mathcal{D}$, in which case, f_j *interpolates* all the data set. If

$$1 = F_j N_{\mathcal{D}} F_j^t = \frac{1}{q}\sum_{i=1}^{q}\|\nabla f_j(p_i)\|^2$$

as well, then F_j has to be a row of the minimizer matrix $\hat{F}$ of (8)–(9) when such a minimizer exists. In Appendix B, we analyze the existence and uniqueness of the solution and show that if $N_{\mathcal{D}}$ is positive definite and $\hat{F}_1, \ldots, \hat{F}_k$ are the eigenvectors of the symmetric-positive pencil $M_{\mathcal{D}} - \lambda N_{\mathcal{D}}$ corresponding to the least k eigenvalues $0 \le \lambda_1 \le \cdots \le \lambda_k$

$$\hat{F}_i M_{\mathcal{D}} = \lambda_i \hat{F}_i N_{\mathcal{D}} \qquad i = 1, \ldots, k$$

$$\hat{F}_i N_{\mathcal{D}} \hat{F}_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i, j = 1, \ldots k$$

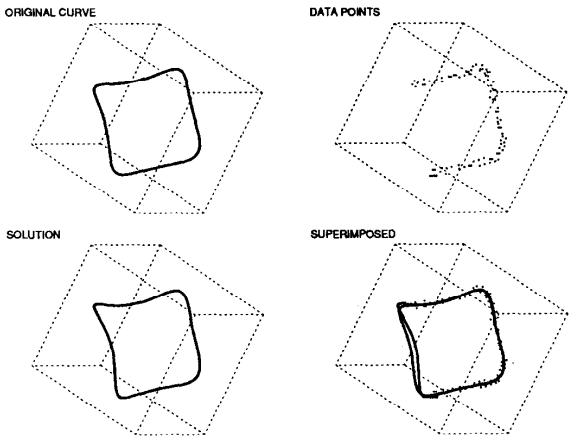


Fig. 8. Generalized eigenvector space curve fit. The solution is the intersection of two unconstrained quadrics. Discontinuities in the curves are errors of the plotting routine.

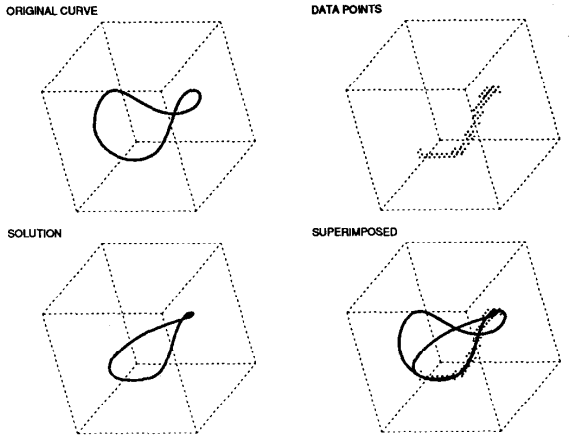


Fig. 10. Generalized eigenvector space curve fit. The solution is the intersection of two unconstrained quadrics. Discontinuities in the curves are errors of the plotting routine.

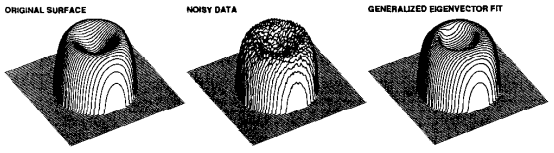


Fig. 9. Fourth-degree algebraic surface fit.

then $\hat{F}$ (the matrix with rows $\hat{F}_1, \ldots, \hat{F}_k$) is a minimizer of (8)–(9).

In general, the matrix $N_{\mathcal{D}}$ is rank deficient, but the problem can be reduced to an equivalent positive-definite one of size equal to the rank of $N_{\mathcal{D}}$. A minimizer exists if and only if the rank of $N_{\mathcal{D}}$ is at least k, and the solution is unique if $k = \operatorname{rank}(N_{\mathcal{D}})$, or if $k < \operatorname{rank}(N_{\mathcal{D}})$ and $\lambda_k < \lambda_{k+1}$. The robustness, or stability, of the solution is also related to how λ_k is separated from λ_{k+1}.

We have already shown examples of generalized eigenvector planar curve fit in Figs. 3 and 4. Fig. 8 shows an example of generalized eigenvector nonplanar space curve fit: the intersection of two cylinders. Fig. 9 shows an example of algebraic fourth-degree surface fit. The original surface is the surface of revolution generated by the curve of Fig. 7, and the solution is the set of zeros of a general fourth-degree polynomial with 35 coefficients. Fig. 10 shows another example of generalized eigenvector space curve fit, which again is the intersection of two surfaces, where the solution fits the data very well, but it is very different from the original curve elsewhere. In general, even if the solution fits the data well, we cannot expect to reconstruct the same original curve or surface, which is the curve or surface of which the data is a sample. Depending on the amount of noise and the extent of the original curve or surface sampled by the data, the generalized eigenvector fit may or may not produce this kind of solution. It is less likely to occur if the reweight procedure and the Levenberg-Marquardt algorithms are used, and the solution is tested after each of the three steps.

VIII. Computational Cost

The complexity of evaluating the matrices $M_{\mathcal{D}}$ and $N_{\mathcal{D}}$ depends on the functions that constitute the vector X. In the case of polynomials when X is the vector of monomials of degree $\leq d$, the elements of $M_{\mathcal{D}}$ are moments of degree $\leq 2d$, and the elements of $N_{\mathcal{D}}$ are integral combinations of at most n moments of degree $\leq 2(d-1)$. For example, for polynomials of degree two in three variables, we have

$$X = (1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2)^t.$$

Other basis vectors, such as the Bernstein basis, are more expensive to evaluate but are potentially more stable [32], [31]. The solution is independent of the basis, however, as we explain in Section IX. We circumvent the stability problem by using centered and scaled monomials, which is also explained in Section IX. The *center* of the data set is its mean value, and the *scale* is the square root of the mean square distance to the center. The computation of the center and scale only requires the moments of degree ≤ 2, which are exactly those moments used for the eigenvector line or planar fit.

Computing the matrices from their definitions requires more operations than computing the vector of moments of degree $\leq 2d$ and then filling the matrices by table lookup. The tables are computed off line. The vector of monomials of degree $\leq 2d$ has $s = \binom{2d+n}{n} = O(h)$ components and can be evaluated by exactly $s - n - 1$ multiplications, where $h = \binom{d+n}{n} = O(d^n)$ is the number of components of X, which is the size of the matrices. It follows that $q(s-n-1)$ multiplications and $(q-1)(s-n-1)$ additions are required to evaluate the moments and then $h(h+1)/2$ operations to fill $M_{\mathcal{D}}$ and at most $nh(h+1)$ operations to fill $N_{\mathcal{D}}$. The total number of operations required to build the matrices is $O(qh + nh^2)$.

An algorithm that computes all the eigenvalues and eigenvectors is given by Golub and Van Loan [42], requiring about $7h^3$ *flops*, where h is the order of the matrices, and a *flop* roughly constitutes the effort of doing a floating point add, a floating point multiply, and a little subscripting. That

algorithm uses the symmetric QR algorithm to compute all the eigenvalues and corresponding eigenvectors. When all the eigenvectors are computed, the symmetric QR algorithm requires about $5h^3$ flops.

Since we only need to compute a few eigenvalues and eigenvectors, we use the alternative methods implemented in EISPACK [68], [36], that is, tridiagonalization, the QR algorithm, and inverse iteration. Tridiagonalization of a symmetric matrix requires about $2/3h^3$ flops. The computation of each eigenvalue using the QR algorithm without computing eigenvectors requires about $5h$ flops per eigenvalue. When the eigenvalues are well separated from each other, as is generally the case in our matrices due to measurement errors and noise, only one inverse iteration is required to compute an eigenvector. Each inverse iteration requires about $4h$ flops. From all this analysis, we conclude that the proposed algorithm requires about $3h^3$ flops. The alternative implicit curve and surface-fitting algorithms all have the same order of complexity [20], [60].

IX. Independence and Invariance

The solution produced by the generalized eigenvector fit method is independent of the basis. In particular, it is independent of the order of the elements of a particular basis. If $Y_1, \ldots, Y_h$ is another basis of the vector space spanned by $X_1, \ldots, X_h$ and we denote $Y = (Y_1, \ldots, Y_h)^t$, then there exists a nonsingular $h \times h$ matrix A such that $X = AY$. Every admissible function can be written as $f(x) = FX = (FA)Y$, and

$$\begin{aligned} FM_{\mathcal{D}}F^t - F\left(\frac{1}{q}\sum_{i=1}^{q} X(p_i)X(p_i)^t\right)F^t \\ = (FA)\left(\frac{1}{q}\sum_{i=1}^{q} Y(p_i)Y(p_i)^t\right)(FA)^t \end{aligned}$$

with the corresponding identity for $N_{\mathcal{D}}$ because, by linearity of differentiation

$$D(FX) = FD(AY) = (FA)DY.$$

It follows that F solves the generalized eigenvector fit problem with respect to the basis X if and only if $[FA]$ solves the same problem but with respect to the basis Y. Since the approximate distance and *a fortiori* the approximate mean square distance are clearly independent of the basis, the minimizer of the approximate mean square distance is independent of the basis as well.

If f is a solution of the generalized eigenvector fit or a minimizer of the approximate mean square distance to the data set $\mathcal{D} = \{p_1, \ldots, p_q\}$, i.e., the curve or surface $Z(f)$ *best fits* the data set $\mathcal{D}$, we want to know for which families of nonsingular transformations T the transformed curve or surface $T^{-1}[Zf] = Z(f \circ T)$ best fits the transformed data set $T^{-1}[\mathcal{D}] = \{T^{-1}(p_1), \ldots, T^{-1}(p_q)\}$.

If the vector space spanned by X is the space of polynomials of maximum degree d, then the solution of the generalized eigenvector fit is invariant with respect to similarity transformations. If $T(x) = Ax + b$ is an nonsingular affine transformation and $f(x)$ is a polynomial, the composition $f(T(x))$ is a polynomial of the same degree whose coefficients are polynomials in A, b, and the coefficients of $f(x)$. If the components of X form a basis of the vector space of polynomials of degree $\leq d$, then so do the components of $Y(x) = X(T(x))$ because the transformation T is nonsingular. It follows that there exists an $h \times h$ nonsingular matrix $T^\star$ whose coefficients are polynomials of degree $\leq d$ in A and b such that $X(T(x)) \equiv T^\star X(x)$ is a polynomial identity (see chapter III, section 4 of [75]). Furthermore, the map $T \mapsto T^\star$ defines a linear representation of the n-dimensional affine group in the h-dimensional general linear group, which is a $1-1$ homomorphism of groups [70] that, in particular, satisfies $(T^{-1})^\star = (T^\star)^{-1}$. For example, if T is a translation in the plane

$$T(x) = \begin{pmatrix} x_1 + b_1 \\ x_2 + b_2 \end{pmatrix}$$

and X is the vector of monomials of degree ≤ 2 in two variables

$$X = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)^t$$

then

$$\begin{aligned} X(T(x)) &= \begin{pmatrix} 1 \\ x_1 + b_1 \\ x_2 + b_2 \\ (x_1 + b_1)^2 \\ (x_1 + b_1)(x_2 + b_2) \\ (x_2 + b_2)^2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ b_1 & 1 & 0 & 0 & 0 & 0 \\ b_2 & 0 & 1 & 0 & 0 & 0 \\ b_2^1 & 2b_1 & 0 & 1 & 0 & 0 \\ b_1b_2 & b_2 & b_1 & 0 & 1 & 0 \\ b_2^2 & 0 & 2b_2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{pmatrix} \\ &= T^\star X(x). \end{aligned}$$

If $f = FX$ is a solution of the generalized eigenvector fit for the data set $\mathcal{D}$ and T is a similarity transformation, that is, $A = \lambda U$, with U orthogonal and λ a nonzero constant, then $g = f \circ T = (FT^\star)X$ is the solution of the same problem for the data set $T^{-1}[\mathcal{D}]$ because

$$\begin{aligned} M_{T^{-1}[\mathcal{D}]} &= \frac{1}{q}\sum_{i=1}^{q} X(T^{-1}(p_i))X(T^{-1}(p_i))^t \\ &= \frac{1}{q}\sum_{i=1}^{q}((T^{-1})^\star X(p_i))((T^{-1})^\star X(p_i))^t) \\ &= (T^\star)^{-1}M_{\mathcal{D}}(T^\star)^{-t} \end{aligned}$$

which implies that

$$(FT^\star)M_{T^{-1}[\mathcal{D}]}(FT^\star)^t = FM_{\mathcal{D}}F^t$$

and with respect to the matrix $N_{\mathcal{D}}$, by the chain rule

$$\begin{aligned} DY &= D(X(T(x)) = (DX)(T(x)) \cdot D(T(x)) \\ &= T^\star\, DX\, A = \lambda T^\star\, DX\, U \end{aligned}$$

which, since U is orthogonal, implies that

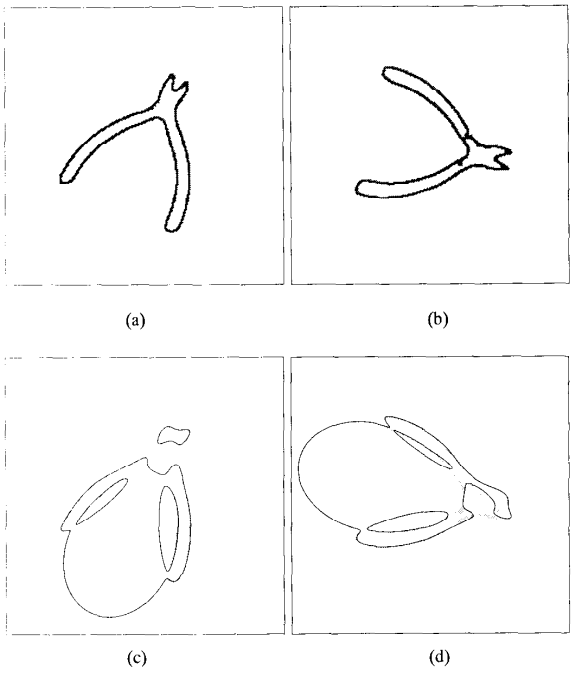


Fig. 11. Invariance to translation and rotation: (a) Data set A; (b) data set B; (c) degree 6 generalized eigenvector fit to data set A; (d) degree 6 generalized eigenvector fit to data set B.

$$N_{T^{-1}[\mathcal{D}]} = \lambda^2 (T^\star)^{-1} N_{\mathcal{D}} (T^\star)^{-t}$$

and so

$$(FT^\star) N_{T^{-1}[\mathcal{D}]} (FT^\star)^t = \lambda^2 \, F N_{\mathcal{D}} F^t.$$

The factor λ^2 in the constraint equation does not change the solution of the problem.

A similar result holds for the minimizer of the approximate mean square distance (7), but we omit the proof, which can easily be reproduced from the previous derivation.

Fig. 11 shows an example of generalized eigenvector fit of sixth-degree algebraic curves to two observations of the same object in different positions. The curves are slightly different because the data sets are as well.

The matrix $M_{\mathcal{D}}$ could be very badly conditioned when X is the vector of monomials of degree $\leq d$. The matrix $N_{\mathcal{D}}$ is generally better conditioned than $M_{\mathcal{D}}$ but not too much. When the matrices are poorly conditioned, the results are not accurate or even totally wrong. Even worse, the eigenvalue computation can diverge. Based on the results of the current section, we would like to find a new basis $Y = AX$ for the polynomials of degree $\leq d$ such that the matrices computed with respect to this new basis are well conditioned. Such a basis has to be a function of the data set. One of such basis is the Bernstein basis [32], [31]. Since the evaluation of the Bernstein basis has a higher complexity than the power basis, we follow an alternative approach that is usually found in the statistical literature [24], [17]. The *center* of the data set is its mean value

$$\overline{p} = \frac{1}{q} \sum_{i=1}^{q} p_i$$

and its *scale* is the square root of the mean square distance to the center

$$\sigma^2 = \frac{1}{q} \sum_{i=1}^{q} \|p_i - \overline{p}\|^2 = \operatorname{trace}\left(\frac{1}{q} \sum_{i=1}^{q} [p_i \; p_i^t] - \overline{p} \, \overline{p}^t \right).$$

If T is the similarity transformation $T(x) = \sigma \cdot x + \overline{p}$, then solving the generalized eigenvector fit or minimizing the approximate mean square distance with respect to the original data set $\mathcal{D}$ is equivalent to solving the corresponding problem using the power basis with respect to the transformed set $T^{-1}[\mathcal{D}]$; this is the proper way to implement this algorithm. First, the center and scale are computed, and then, the data set is transformed and the corresponding fitting problem is solved using the power basis; finally, the coefficients are back-transformed using $T^\star$. It is even better not to backtransform the polynomial g, which is the solution of the transformed problem, and evaluate the solution of the original problem f in two steps

$$\begin{cases} x' & = \frac{1}{\sigma}(x - \overline{p}) \\ f(x) & = g(x') \end{cases}$$

when such an evaluation becomes necessary.

X. The Reweight Procedure

Let $f = \phi_\alpha$ and $w_1, \ldots, w_q$ be positive numbers. Let us call

$$\xi^2_{\mathcal{D},w}(\alpha) = \frac{1}{q} \sum_{i=1}^{q} w_i \|f(p_i)\|^2 \tag{10}$$

the *weighted mean square error*. For planar curves and surfaces $k = 1$ and $w_i = 1/\|\nabla f(p_i)\|^2$, the weighted mean square error reduces to the approximate mean square distance. The point p_i is given a heavy weight only if it is close to a singular point of $Z(f)$. Note that if $\hat{f}$ is a minimizer of the approximate mean square distance constrained by (5) and p_i is close to a singular point of $Z(\hat{f})$, then both $\|\hat{f}(p_i)\|^2 \approx 0$ and $\|\nabla \hat{f}(p_i)\|^2 \approx 0$ and the contributions of p_i to both the mean square error (7) and the constraint (5) are negligible. Minimizing the mean square error instead of the approximate mean square distance is like fitting a curve or surface to the data set but without considering the points close to singularities of $Z(\hat{f})$. If the minimizer of the approximate mean square distance has singular points, we cannot expect them to be well approximated by the set of zeros of the curve or surface solution of the generalized eigenvector fit problem.

In general, we take

$$w_i(\alpha) = \operatorname{trace}([Df(p_i) Df(p_i)^t]^{-1})$$

and based on the same arguments used in Section V, we

```
procedure Reweight (F, D)
  F' := F
  do
     F  := F'
     w  := w(F)
     F' := minimizer of  trace(F^t M_{D,w} F)
           constrained by  F^t N_{D,w} F = I_k
  while Δ²_D(F') < (1 − ε)Δ²_D(F)
  if Δ²_D(F') < Δ²_D(F) then
     return (F')
  else
     return (F)
```

Fig. 12. Reweight procedure for the generalized eigenvector fit.

impose the constraint

$$\frac{1}{q}\sum_{i=1}^{q} w_i[Df(p_i)Df(p_i)^t] = I_k \tag{11}$$

without modifying the set of zeros of the minimizer of the approximate mean square distance.

Now, if we keep $w = (w_1, \ldots, w_q)^t$ fixed and we stay within the linear model $f(x) = FX(x)$, the minimization of (10) constrained by (11) is a generalized eigenproblem, as in the previous section. In this case, we minimize $\text{trace}(FM_{\mathcal{D},w}F^t)$ constrained by $FN_{\mathcal{D},w}F^t = 1$, where

$$M_{\mathcal{D},w} = \frac{1}{q}\sum_{i=1}^{q} w_i\,[X(p_i)X(p_i)^t]$$

$$N_{\mathcal{D},w} = \frac{1}{q}\sum_{i=1}^{q} w_i\,[DX(p_i)DX(p_i)^t]$$

If $\hat{f}$ is a minimizer of the approximate mean square distance and the weighs $w_1, \ldots, w_q$ are close to $w_1(\hat{f}), \ldots, w_q(\hat{f})$, by continuity, the minimizer of the weighted linear problem is close to $\hat{f}$. This property suggests the *reweight procedure*, which is described in Fig. 12, where ϵ is a small positive constant that controls the stopping of the loop.

In the linear model, the initial value of F will be the solution produced by the generalized eigenvector fit with uniform weights, as is the case of the previous section. In a practical implementation, and in order to save time, the reweight procedure would be called only if this initial value of F does not pass a *goodness of fit* test. At each iteration, we solve a generalized eigenproblem and then recompute the weights. We continue doing this while the approximate mean square distance decreases. Then, if the value of F returned by the reweight procedure does not pass the same goodness of fit test, we call the Levenberg-Marquardt algorithm.

The reweight procedure is similar in spirit to Sampson's algorithm [64] and those iterative weighted least squares algorithms that appear in the regression literature [61]. There is no certainty of convergence, however, and according to Sampson, the system of equations is so complex that it would be extremely difficult to determine the conditions under which such a scheme converges [47]. This statement agrees with what we have observed during the course of this study.

We have already shown in Fig. 3 how the reweight procedure improves the result of the generalized eigenvector fit, even though the final result would not be accepted by the goodness of fit test. In Fig. 13, we show how the reweight procedure improves the fitting curve in a case where the result provides an acceptable approximation of the data.

(a) (b)

Fig. 13. Reweight procedure improving generalized eigenvector fit: (a) Generalized eigenvector fit; (b) after reweight procedure.

XI. Other Families of Curves and Surfaces

In this section, we consider several potential applications of the methods introduced in this paper.

In certain cases, such as in the family of cylinders or superquadrics, the parameters can be divided in two groups: *shape* and *positional* parameters. The radius of a cylinder is a shape parameter, and all the other parameters, which describe its axis, are positional parameters. In the family of polynomials of a given maximum degree, all the parameters are shape parameters because the composition of a polynomial with a nonsingular affine transformation is a polynomial of the same degree.

It is particularly important to consider the case in which all the parameters are positional parameters: the case of a family of compositions of a fixed map g with a parameterized family of transformations $\mathcal{G}$

$$\mathcal{F} = \{f : \exists\, T \in \mathcal{G} f(x) \equiv g(T(x))\}$$

for its applications to object recognition and position estimation.

A. Transformations of a Curve or Surface

A family of transformations, e.g., rigid body, affine, or projective transformations, can be given in parametric form

$$\mathcal{G} = \{T_\alpha : \alpha \in IR^r\}$$

in which case the parameterization of the admissible functions is

$$\phi(\alpha, x) = g(T_\alpha(x))$$

where $g(x)$ is a fixed map whose set of zeros we will call the *model in standard position*. Explicit parameterizations of some of these families are discussed below in this section. The problem of fitting a member of the family $\mathcal{F} = \{f : \exists\alpha f(x) \equiv g(T_\alpha(x))\}$ to a set of data points $\mathcal{D} = \{p_1, \ldots, p_q\}$ is position estimation or *generalized template matching*; we assume that the data belong to a single object and that this object is an observation of the model not necessarily in the standard position and possibly partially occluded. We minimize $\Delta^2_{\mathcal{D}}(\alpha)$,

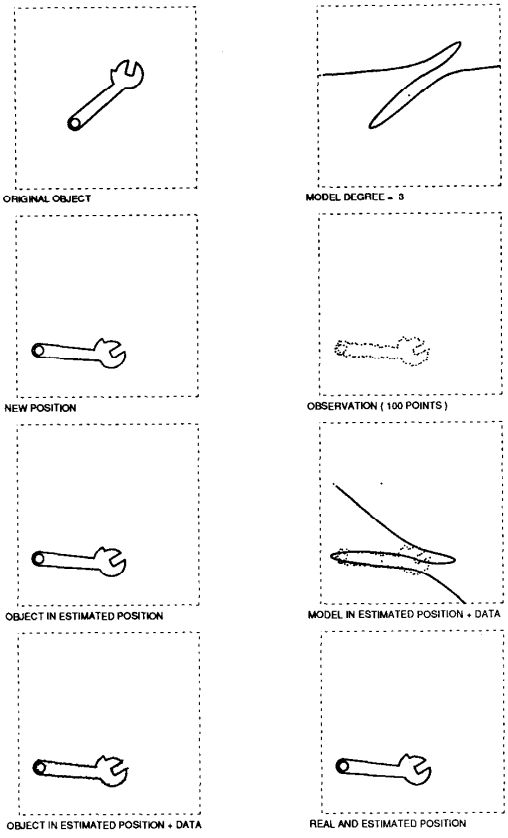


Fig. 14. Simple position estimation.

which is the approximate mean square distance from the data points to the model $Z(g)$ transformed to the position and orientation defined by the transformation T_α

$$\begin{aligned} Z(g \circ T_\alpha) &= \{x : g(T_\alpha(x)) = 0\} \\ &= \{T_\alpha{}^{-1}(y) : g(y) = 0\} \\ &= T_\alpha^{-1}[Z(g)]. \end{aligned}$$

This method was used by Cooper *et al.* [21] to locate surface matching curves in the context of the stereo problem. Fig. 14 shows a simple example of planar curve position estimation within this framework. How to choose initial estimates for the minimization of the approximate mean square distance is the remaining problem of this formulation, particularly when we have to deal with very complex objects or occlusions, and it is the subject of our current research. In Section XIII, we describe how we intend to attack this problem through the concept of *interest regions.*

B. Projection of Space Curves Onto the Plane

The methods for the minimization of the approximate mean square distance introduced in this paper can be used to improve Poncc and Kriegman's algorithm [59] for object recognition and positioning of 3-D objects from image contours.

In their formulation, object models consist of collections of parametric surface patches and their intersection curves. The image contours considered are the projections of surface discontinuities and occluding contours. They use elimination theory [28], [41], [51], [63], [66], [18] for constructing the implicit equation of the image contours of an object observed under orthographic, weak perspective, or perspective projection. The equation is parameterized by the position and orientation of the object with respect to the observer.

Note that object models defined by implicit curves and surfaces can be handled using the same techniques. For example, under orthographic projection, the implicit equation of the projection of an algebraic curve $Z(f)$ onto the plane can be computed, eliminating the variable x_3 from the pair $(f_1(x), f_2(x))$, and since the occluding contour of a surface $Z(f)$ is the curve defined as the intersection of $Z(f)$ with $Z(\partial f/\partial x_3)$, the projection of the occluding contour is in this case the *discriminant* of f with respect to z_3 (see chapter I, section 9 of [75]).

They use more complex iterative methods to compute the distance from a point to the hypothesized curve than the approximate distance, and so, it becomes much more expensive to minimize their approximation to the mean square distance than in our case.

C. Parameterizations of Some Transformation Groups

The general rigid body transformation is $T(x) = Qx + b$, where Q is a rotation matrix, which is an orthogonal matrix of unitary determinant, and b is an n-dimensional vector. The general affine transformation can be written, as usual, as $T(x) = Ax + b$, where A is a $n \times n$ nonsingular matrix, and b is an n-dimensional vector. Alternatively, we can write it as $T(x) = LQx + b$, where L is a nonsingular lower triangular matrix, and Q is a rotation matrix.

The usual parametcrization of a general rotation matrix is trigonometric $Q(\theta)$, which is the product of the following three matrices:

$$\begin{pmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{pmatrix}, \text{and}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{pmatrix}$$

where $\theta = (\theta_1, \theta_2, \theta_3)$ are the so called Euler angles. Since we have to iterate on the parameters, we propose the alternative rational parameterization, originally due to Cayley (see chapter II of [76]), which is valid in any dimension.

A square matrix Q is called *exceptional* if

$$|I + Q| = 0$$

and *nonexceptional* otherwise, where I is the identity matrix, that is, a matrix is exceptional if it has -1 as an eigenvalue.

If the matrix Q is nonexceptional, let us consider the matrix

$$U = (I-Q)(I+Q)^{-1} = (I+Q)^{-1}(I-Q).$$

The matrix U so defined is nonexceptional as well because

$$\begin{aligned}|I+U| &= \left|(I+Q)^{-1}((I+Q)+(I-Q))\right| \\ &= 2|I+Q|^{-1}\end{aligned}$$

and the matrix Q can be recovered from U in the same way because

$$\begin{aligned}I-U &= \left(((I+Q)-(I-Q))(I+Q)^{-1}\right) \\ &= 2Q(I+Q)^{-1} \\ I+U &= \left(((I+Q)+(I-Q))(I+Q)^{-1}\right) \\ &= 2I(I+Q)^{-1} \\ (I-U)(I+U)^{-1} &= \left(2Q(I+Q)^{-1}\right)\left(2I(I+Q)^{-1}\right)^{-1} \\ &= Q.\end{aligned}$$

Furthermore, another simple algebraic manipulation shows that Q is a rotation if and only if U is skew symmetric.

For $n=2$, a skew-symmetric matrix can be written as

$$U = \begin{pmatrix} 0 & u \\ -u & 0 \end{pmatrix}$$

and no real 2×2 skew-symmetric matrix is exceptional because

$$|I+U| = 1+u^2 \geq 1.$$

The rational parameterization of the 2×2 nonexceptional rotation matrices is defined by the map $\mathbb{R}^2 \rightarrow O(2)$ given by

$$\begin{aligned}Q(u) &= \begin{pmatrix} 1 & -u \\ u & 1 \end{pmatrix}\begin{pmatrix} 1 & u \\ -u & 1 \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \frac{1-u_2}{1+u_2} & \frac{-2u}{1+u_2} \\ \frac{2u}{1+u_2} & \frac{1-u_2}{1+u_2} \end{pmatrix}\end{aligned}$$

The only exceptional 2-D rotation is the matrix

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \tag{12}$$

which corresponds to a rotation of π radians.

For $n=3$, we can write the general skew-symmetric matrix in the following way:

$$U = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}$$

because if $u=(u_1,u_2,u_3)^t$ and v are two 3-D vectors, then Uv is equal to the vector product $u\times v$. Again, no skew-symmetric 3×3 matrix is exceptional because $|I+U| = 1+u_1^2+u_2^2+u_3^2 geq 1$, and the only exceptional rotation matrices are those corresponding to rotations of π radians. The rational parameterization of the 3×3 rotation matrices is defined by the map $\mathbb{R}^3 \rightarrow O(3)$ given by

$$Q(u) = \begin{pmatrix} \frac{1+u_1^2-u_2^2-u_3^2}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_1u_2-u_3)}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_1u_3+u_2)}{1+u_1^2+u_2^2+u_3^2} \\ \frac{2(u_1u_2+u_3)}{1+u_1^2+u_2^2+u_3^2} & \frac{1-u_1^2+u_2^2-u_3^2}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_2u_3-u_1)}{1+u_1^2+u_2^2+u_3^2} \\ \frac{2(u_1u_3-u_2)}{1+u_1^2+u_2^2+u_3^2} & \frac{2(u_2u_3+u_1)}{1+u_1^2+u_2^2+u_3^2} & \frac{1-u_1^2-u_2^2+u_3^2}{1+u_1^2+u_2^2+u_3^2} \end{pmatrix} \tag{13}$$

Note that this function maps a neighborhood of the origin $u=0$ continuously onto a neighborhood of the identity matrix $Q(0)=I$. Clearly, if Q_0 is a rotation, then, the map $u \mapsto Q_0Q(u)$ maps a neighborhood of the origin onto a neighborhood of the matrix Q_0, and in this way, we can deal with the exceptional orthogonal matrices. Also note that the three parameters have explicit geometrical meaning. If $u_2=u_3=0$, the matrix $Q(u)$ becomes

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1-u_1^2}{1+u_1^2} & \frac{-2u_1}{1+u_1^2} \\ 0 & \frac{2u_1}{1+u_1^2} & \frac{1-u_1^2}{1+u_1^2} \end{pmatrix}$$

which is a rotation about the x_1 axis. Similarly, $Q(u)$ is a rotation about the x_2 axis if $u_1=u_3=0$, and it is a rotation about the x_3 axis if $u_1=u_2=0$. More generally, if $v(t)$ describes the trajectory of a point that rotates with constant angular velocity, the motion equations can be written as $\dot{v} = u\times v = Uv$, where $\dot{v}$ is the time derivative of $v(t)$. In the language of Lie groups, the skew-symmetric matrices are the *infinitesimal generators* of the group of rotations.

D. Cylinders, Algebraic Curves, and Surfaces

The implicit equations of a straight line can be written as

$$\begin{cases} Q_1^t x - \alpha_4 = 0 \\ Q_2^t x = 0 \end{cases}$$

where Q_1, Q_2, Q_3 are the columns of a general rotation matrix $Q(\alpha_1,\alpha_2,\alpha_3)$ parameterized as in (13), with $\alpha_1,\alpha_2,\alpha_3$ real parameters, in which case the general implicit representation of a cylinder is given as the set of zeros of

$$\begin{aligned}&(Q_1^t x-\alpha_4)^2 + (Q_2^t x)^2 - \alpha_5^2 \\ &= x^t(I-Q_3Q_3^t)x - 2\alpha_4 Q_1^t x + \alpha_4^2 - \alpha_5^2\end{aligned} \tag{14}$$

which is the set of points at a distance $|\alpha_5|$ from the straight line. By homogeneity, the set of zeros of (14) does not change if we multiply it by the constant $\kappa^2 = (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)^2 \neq 0$ so that the cylinder can also be represented as the set of zeros of

$$f(x) = x^t(\kappa^2 I - q_3q_3^t)x - 2\alpha_4\kappa q_1^t x + \kappa^2(\alpha_4^2-\alpha_5^2), \tag{15}$$

where

$$q_1 = \kappa Q_1 = \begin{pmatrix} 1+\alpha_1^2-\alpha_2^2-\alpha_3^2 \\ 2(\alpha_1\alpha_2+\alpha_3) \\ 2(\alpha_1\alpha_3-\alpha_2) \end{pmatrix}$$

and

$$q_3 = \kappa Q_2 = \begin{pmatrix} 2(\alpha_1\alpha_3+\alpha_2) \\ 2(\alpha_2\alpha_3-\alpha_1) \\ 1-\alpha_1^2-\alpha_2^2+\alpha_3^2 \end{pmatrix}$$

This representation is particularly attractive because the coefficients of $f(x)$ are *polynomials* in the five unconstrained

parameters $\alpha_1, \ldots, \alpha_5$ and are therefore much less expensive to evaluate than the trigonometric functions. Explicitly, $f(x) = F(\alpha) \cdot X(x)$, where

$$F(\alpha) = \begin{pmatrix} (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_4^2-\alpha_5^2) \\ -2\alpha_4\left((1+\alpha_1^2)^2-(\alpha_2^2+\alpha_3^2)^2\right) \\ -4\alpha_4(1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_1\alpha_2+\alpha_3) \\ -4\alpha_4(1+\alpha_1^2+\alpha_2^2+\alpha_3^2)(\alpha_1\alpha_3-\alpha_2) \\ (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)^2-4(\alpha_1\alpha_3+\alpha_2)^2 \\ -8(\alpha_1\alpha_3+\alpha_2)(\alpha_2\alpha_3-\alpha_1) \\ -4(\alpha_1\alpha_3+\alpha_2)(1-\alpha_1^2-\alpha_2^2+\alpha_3^2) \\ (1+\alpha_1^2+\alpha_2^2+\alpha_3^2)-4(\alpha_2\alpha_3-\alpha_1) \\ -4(\alpha_2\alpha_3-\alpha_2)(1-\alpha_1^2-\alpha_2^2+\alpha_3^2) \\ 4(1+\alpha_1^2+\alpha_2^2)(1+\alpha_3^2)-4 \end{pmatrix}^t$$

and

$$X(x) = (1x_1x_2x_3x_1^2x_1x_2x_1x_3x_2^2x_2x^3x_3^2)^t.$$

The same procedure that we applied above (the multiplication of the implicit equation of the cylinder by a power of κ to convert the rational parameterization of the coefficients into a polynomial one) can be applied to the composition of a fixed polynomial with a rigid body transformation to obtain a polynomial parameterization of all the members of the family, even if some shape parameters are present, as in the cylinder.

E. Superquadrics

According to Solina [69], superquadrics were discovered by the Danish designer Hein [37] as an extension of basic quadric surfaces and solids. They have been used as primitives for shape representation in computer graphics by Barr [4], in computer-aided design by Elliot [30], and in computer vision by Pentland [58], Bajcsy and Solina [3], Solina [69] and Boult and Gross [15], [16], [44].

Barr divides the superquadrics into superellipsoids, superhyperboloids, and supertoroids. We will only consider the case of superellipsoids here as an example. The usual representation of a superellipsoid is the set of zeros of $f(x) = g(x) - 1$, where

$$g(x) = \left(\left|\frac{u^t x}{a_1} - b_1\right|^{\frac{2}{\epsilon_1}} + \left|\frac{v^t x}{a_2} - b_2\right|^{\frac{2}{\epsilon_1}}\right)^{\frac{\epsilon_1}{\epsilon_2}} + \left|\frac{w^t x}{a_3} - b_3\right|^{\frac{2}{\epsilon_1}}$$

and $[uvw]$ is a general rotation matrix parameterized by the three Euler angles $\theta_1, \theta_2, \theta_3$, and $a_1, a_2, a_3, b_1, b_2, b_3, \epsilon_1, \epsilon_2$ are unconstrained parameters. Solina [69] proposes an alternative parameterization with

$$f(x) = \sqrt{a_1a_2a_3}(|g(x)|^{\epsilon_2} - 1)$$

to overcome the bias problem associated with the minimization of the mean square error, and Gross and Boult [44] compare the performance of these two representations with other two, where one of them is proposed by themselves.

We propose the following parameterization (a generalization of the parameterization of cylinders (15)), which simplifies the computation of the partial derivatives with respect to α, which is required by the Levenson-Marquardt algorithm

$$\phi(\alpha, x) = \left(\alpha_7^2(q_1^t x - \alpha_4\kappa)^{2\alpha_{10}} + \alpha_8^2(q_2^t x - \alpha_5\kappa)^{2\alpha_{10}}\right)^{\frac{\alpha_{11}}{\alpha_{10}}} + \alpha_9^2(q_3^t x - \alpha_6\kappa)^{2\alpha_{11}} - \kappa^{2\alpha_{11}}$$

where q_1, q_2, and q_3 are the three columns of the parameterized rotation matrix (13) multiplied by κ^2. Superhyperboloids and supertoroids can be parameterized in a similar way.

XII. Related Work on Implicit Curve and Surface Fitting

According to Duda and Hart [29], the earliest work on fitting a line to a set of points was probably motivated by the work of the experimental scientist; the easiest way to explain a set of observations is to relate the dependent and independent variables by means of the equation of a straight line. Minimum-squared-error line fitting (see section 9.2.1 of [29]) and eigenvector line fitting (see section 9.2.2 of [29]) are the two classical solutions to this classical problem. With our notation, in both cases, the mean square error (6) is minimized, with $F = (F_1, F_2, F_3)$ and $X = (1, x_1, x_2)^t$. In the first case, the linear constraint $F_3 = 1$ is imposed, whereas the quadratic constraint $F_2^2 + F_3^2 = 1$ is imposed in the second case. Early work on eigenvector line fitting can be traced back to Pearson [57] and Hotelling [46] on principal components analysis. Chapter 1 of Jollife [48] gives a brief history of principal components analysis.

Pratt [60] affirms that there appears to be relatively little written about fitting planar algebraic curves to data [55], [54], [8], [1], [72], [22], [23], [14], [40], [64], [20] and none whatsoever of least-squares fitting of nonplanar algebraic surfaces. We can add to this statement that we have been unable to locate any previous work on fitting space curves defined by implicit functions, except for Ponce and Kriegman [59], who fit the *projection* of a space curve to 2-D data points and only a few references on fitting quadric surfaces to data [38], [45], [34], [19], [9], [10]. However, there is an extensive literature on fitting parametric curves and surfaces to scattered data, for example [26], [35], [39], [50], [56].

The first extensions of the line fitting algorithms concentrated on fitting conics to data minimizing the mean square error. Since the implicit equations are homogeneous, a constraint has to be imposed on the coefficients to obtain a nontrivial solution, and either linear or quadratic constraints were considered. With our notation, a conic is the set of zeros of a polynomial $f(x) = FX$, where $F = (F_1, \ldots, F_6)$, and $X = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_2^2)^t$. Biggerstaff [8], Albano [1], and Cooper and Yalabik [22], [23] impose the constraint $F_1 = 1$. This constraint has the problem that no conic that passes through the origin satisfies it, and therefore, it cannot be a solution of the problem. Paton [55], [54] imposes the constraint $F_1^2 + \cdots + F_6^2 = 1$, and Gnanadesikan [40] uses $F_2^2 + \cdots + F_6^2 = 1$. These two constraints are not invariant with respect to translation and rotation of the data, whereas Bookstein's constraint [14] $F_4^2 + F_5^2/2 + F_6^2$ is. However, the previous two quadratic constraints allow the solution to have $F_4 = F_5 = F_6 = 0$, which is a straight line, whereas Bookstein

does not. Pratt [60] shows that this is a great disadvantage and proposes a new quadratic constraint for the case of circles. Cernuschi–Frias [19] derives a quadratic constraint that is invariant under the action of the Euclidean group for quadric surfaces, generalizing Bookstein's constraint.

In general, linear constraints turn the minimization of the mean square error into a linear regression problem, whereas quadratic constraints convert it into an eigenvalue problem. It is important to note that in all the previous cases, the constraints are independent of the data. They have been chosen beforehand. In our generalized eigenvector fit, the quadratic constraint is a function of the data.

More recently, Chen [20] fitted general linearly parameterized planar curves to data. With our notation, an admissible function is $f(x) = FX = F_1X_1 + \cdots F_hX_h$. He imposes the constraint $F_h = 1$, turning the minimization into a linear regression problem, and he solves the normal equations with the QR algorithm (see chapter 6 of [42]). This constraint looks arbitrary because it rules out all the curves with $F_h = 0$ or close to zero. A different ordering of the basis vector would produce a different solution. In the case of fitting circles to data shown as an application in Chen's paper, where $X = (1, x_1, x_2, x_1^2 + x_2^2)^t$, a straight line cannot be a valid solution as in Bookstein's algorithm, and the same kind of behavior has to be expected.

Chen apparently developed his algorithm independently of the earlier work of Pratt [60], whose *simple fit* algorithm is very close to Chen's, except that he solves the ordering problem in a clever way. Pratt presents his algorithm for algebraic curves and surfaces, but it can be formulated for the general linear case with the same effort. Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be the set of data points. Let us first assume that $q = h - 1$, where h is the number of elements of X, and the $h \times q$ *design matrix*

$$X_{\mathcal{D}} = [X(p_1) \cdots X(p_q)]$$

has maximal rank $h - 1$. The function

$$f(x) = \det([X_{\mathcal{D}} X])$$

is a linear combination of the elements of X, with the coefficients being the cofactors of the elements of the hth column. Note that for every $i = 1, \ldots, h - 1$

$$f(p_i) = \det([X_{\mathcal{D}} X(p_i)]) = 0$$

because the matrix has two identical columns. The function $f(x)$ *interpolates* the data points. Pratt calls this technique, which generalizes the classical Vandermonde determinant (see chapter II of [25]), *exact fit*. The coefficients can be efficiently computed by first triangularizing $X_{\mathcal{D}}$ via column operations at cost $O(h^3)$ and then computing the cofactors at an additional cost $O(h^2)$. In the general case, he applies the Cholesky decomposition to the square matrix $X_{\mathcal{D}} X_{\mathcal{D}}^t = qM_{\mathcal{D}}$, obtaining the unique square lower triangular matrix L with positive diagonal elements such that $X_{\mathcal{D}} X_{\mathcal{D}}^t = LL^t$, and then, he deletes the last column of L and treats the result as though it were the $h \times (h-1)$ matrix $X_{\mathcal{D}}$ of the exact fit case. In this procedure, the coefficient of X_h is never zero and corresponds to the constraint $F_h = 1$, producing the same solution as Chen's algorithm. The way Pratt overcomes the ordering problem is by performing the Cholesky decomposition with full pivoting, permuting the elements of X during the decomposition. No one coefficient is singled out as having to be nonzero. Pratt calls this procedure *simple fit*.

With respect to iterative methods, Sampson [64] introduced a reweight procedure to improve Bookstein's algorithm in the case of "very scattered" data. He explains that this reweighting scheme does not necessarily converge, but he does not make use of further optimization techniques.

Solina [69] and Gross and Boult [44] fit superquadrics [4] to data minimizing the mean square error (7) using the Levenberg-Marquardt algorithm.

Ponce and Kriegman [59] introduce two iterative methods for estimating the distance from a point to an implicit curve and then use the Levenberg-Marquardt algorithm to minimize the sum of the squares of these estimates. Both methods are obviously more expensive than minimizing the approximate mean square distance.

XIII. Interest Regions as Features for Object Recognition

Our goal is to use the above procedures to develop techniques for recognizing objects in such hostile environments as, for example, a bin of parts or in a cluttered environment. Recognition will be based on matching small regions of observed data with parts of known models. The small regions will be assumed to constitute *unoccluded* observations of the corresponding parts of the models. If an object is modeled as a collection of geometrically related subobjects, the problem is reduced to finding the best collection of pairings between model subobjects and regions of observed data that satisfies the same set of global constraints.

In order to speed up the matching process, we need a procedure to chose interest regions of a model or data set and generalized features that will be used as the primary source of information in the matching process. The fact that implicit curves and surfaces can represent several smooth patches at once, lets us generalize the concept of high curvature points as features for recognition.

We define an *interest region* of degree d and radius r as the data within a circular or spherical (rotation invariant) window of fixed radius r centered at a data or model point, which is well approximated by a dth degree algebraic curve or surface with respect to the chosen goodness of fit criterion; it is not well approximated by any lower degree curve or surface, and it is *stable* with respect to small displacements of the window center. Fig. 15 shows a fourth-degree interest region and radius equal to 25 pixels in a simple planar example. We can see that around the tip of the pliers, the fit is almost independent of the position of the center of the circle, and a third-degree curve does not provide a good fit in either case. The representation of one object as a collection of subobjects has been used before, but the subobjects have been simpler, such as quadric patches. For 2-D edge images, the interest regions allow us to use the information provided not only by the silhouette but by the internal boundary structure as well. The interest regions are

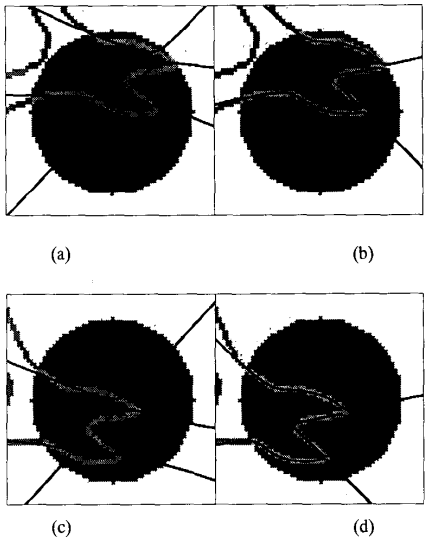


Fig. 15. Interest region of degree 4 and a 25-pixel radius: (a) Poor third-degree fit in the first position; (b) good fourth degree fit in the first position; (c) poor third-degree fit in the second position; (d) good fourth-degree fit in the second position.

subregions with a relatively complex structure, i.e., regions that can only be represented with higher d.

Now, we sketch how a matching process could work based on these interest regions. This is the subject of our current research, and a detailed study will follow this paper.

All the interest regions of the third and fourth degree, and a finite number of radii, will be precomputed for the known models and organized into a database. This database will be indexed by a finite number of algebraic invariants of the approximating polynomial. After finding an interest region in the data set, the matching process will start by computing the algebraic invariants of the stable polynomial of degree d with respect to rigid body transformations and finding an appropriate match in the database. Then, a rotation and translation, which would transform the model polynomial to the estimated one, will be hypothesized, based on tensor algebra techniques. This rigid-body transformation will be used as an initial estimate for the methods of Section XI-A and, after iterative improvement, will be tested against a larger subset of data.

XIV. A Segmentation Algorithm

As an application of the ideas introduced in preceding sections, we have implemented two versions of the same segmentation algorithm. The first one segments planar edge maps into algebraic planar curve patches, and the second one segments range images into algebraic surface patches. In both cases, the maximum degree is a parameter that can be chosen by the user, but due to numerical and complexity considerations, it is not appropriate to use patches of a degree higher than four or five. There is a straightforward extension of these algorithms to the segmentation of space edge maps, which are composed of surface discontinuities, surface normal discontinuities, occluding boundaries, or lines of curvature, into algebraic space curve patches, which we are currently implementing.

These algorithms are partially based on Besl and Jain's variable-order surface fitting algorithm [7], [6] and Silverman and Cooper's surface estimation clustering algorithm [67], and they are also related to Chen's planar curve reconstruction algorithm [20].

Both versions have the same structure with minor differences at the implementation level. The basic building blocks are noise variance estimation, region growing, and merging.

Our philosophy is that this segmentation is most useful when the appropriate segmentation is well defined, i.e., when there are range or surface normal discontinuities between regions, each of which is well represented by a single polynomial.

A. Noise Variance Estimation

Since the algorithm follows an *hypothesize and test* approach, tests for accepting or rejecting a hypothesized curve or surface as a good approximation for a given set of data points have to be chosen. In addition, good subsets of the data set to start the region-growing process (the *seeds*) have to be located. These two subjects are closely related.

Our tests are based on modeling a smooth region of the data set as samples of an implicit curve or surface plus additive perturbations in the orthogonal direction to the curve or surface. These orthogonal perturbations are assumed to be independent Gaussian random variables of zero mean. The square of the noise variance at one data point $\hat{\sigma}(x)$ is estimated by fitting a straight line or plane to the data in a small neighborhood of the point, which in our implementation is a circle or ball of a radius equal to a few pixels, using the eigenvector fit method and then computing the approximate mean square distance to the fitted line or plane. This estimator can be biased if the curvature of the curve or surface is large at the point.

Although a much better method to estimate the noise variance would be to fit a circle or ellipsoid to the data in a neighborhood of the point and then measure the approximate mean square distance to this curve or surface, we have experienced very good results with the former method.

The first step of the algorithm is to estimate the square of the noise variance at every data point and to store these values in an array for latter usage. In the case of surfaces, we only compute the estimated noise variances on a subsampled image to save computational time.

The second step of the algorithm is to build a histogram of the square noise variance. The data points with square noise variance in the top 10% of the histogram are marked as outliers. These points correspond to mixed regions, corners, or surface normal discontinuities. All the remaining points are left available to start growing regions from the small neighborhoods used to estimate their noise variances because a straight line or plane well approximates the data in that neighborhood.

B. Goodness of Fit Tests

During the region-growing process, curves or surfaces are

hypothesized as good approximations for a given subset of data. These hypotheses have to be tested because the criterion used for fitting is different from the desired one. We have chosen the thresholds used to test the validity of a hypothesized curve or surface approximation for a set of data points as functions of the noise variance estimates for the individual points of the set instead of choosing global thresholds for the full data set, as in [7]. There are two reasons for this. In the first place, we have observed that in the case of surfaces, the point noise variance estimated with the method described in the previous section is nonstationary, varying slowly according to the angle of the normal with respect to the vertical, that is, the viewing direction. In the second place, local thresholds allow parallel implementations of the algorithms because distant regions become independent of each other. The data set can be spatially divided into blocks of approximately the same size, the region growing algorithm can be run in parallel for each block, and then pairs of neighboring blocks can also be merged in parallel. A pyramid architecture can be used to implement such an algorithm. The two implementations that we present in this paper are sequential and are only intended to show the usefulness of the previous fitting techniques. We will implement the corresponding parallel versions in the near future.

Let $\mathcal{S} = \{p_1, \ldots, p_q\}$ be a set of data points, and let α be a parameter vector that defines an implicit curve or surface.

The *mean noise variance estimate* on the set $\mathcal{S}$ will be denoted

$$\hat{\sigma}_{\mathcal{S}}^2 = \frac{1}{q} \sum_{i=1}^{q} \hat{\sigma}(p_i)$$

where $\hat{\sigma}(p_i)$ is the noise variance at the point p_i estimated with the method of the previous section. The *maximum approximate square distance* from the set $\mathcal{S}$ to the curve or surface defined by α will be denoted

$$\delta_{\mathcal{S}}^2(\alpha) = \max_{1 \le i \le q} \delta(\alpha, p_i)^2.$$

Two tests are performed on the parameter vector α to decide whether or not to accept the curve or surface defined by α as a good approximation of the set $\mathcal{S}$. The first test is related to the χ^2 statistic. The parameter vector α passes the first test if the approximate mean square distance satisfies the inequalities

$$\epsilon_1 \, \hat{\sigma}_{\mathcal{S}}^2 < \Delta_{\mathcal{D}}^2(\alpha) < \epsilon_2 \, \hat{\sigma}_{\mathcal{S}}^2$$

where $0 < \epsilon_1 < 1 < \epsilon_2$ are test constants. If α satisfies the first test, then we perform the second test. The parameter vector α passes the second test if

$$\delta_{\mathcal{S}}^2(\alpha) < \epsilon_3 \Delta_{\mathcal{D}}^2(\alpha)$$

where $\epsilon_3 > 1$ is another test constant.

If the parameter vector α does not satisfy the first test with $\Delta_{\mathcal{D}}^2(\alpha) \le \epsilon_1 \, \hat{\sigma}_{\mathcal{S}}^2$, that is, with the approximate mean square distance to small, the curve or surface defined by α is *overapproximating* the data set, and the hypothesis has to be rejected, e.g., if we try to test a pair of very close parallel lines as an approximation of a noisy straight line segment. In a variable-order algorithm, this means that the order has to be *decreased*. If the parameter vector α does not satisfy the first test with $\epsilon_1 \, \hat{\sigma}_{\mathcal{S}}^2 \le \Delta_{\mathcal{D}}^2(\alpha)$, that is, with the approximate mean square distance to large, too many points of $\mathcal{S}$ are too far away from the curve or surface defined by α, and it cannot possibly pass the second test. The hypothesis has to be rejected. Finally, our goal is to accept the parameter α only if the curve or surface defined by α approximates every point of $\mathcal{S}$ well, that is, the maximum approximate distance is not to large with respect to the variance estimate for the set. The second test takes care of this situation.

C. Region Growing

The basic structure of the variable-order region growing algorithm can be described as follows. An increasing sequence $\mathcal{F}_1 \subseteq \cdots \subseteq \mathcal{F}_{order_{\mathrm{MAX}}}$ of families of functions is given. A *region* is a data structure $\mathcal{R} = (\mathcal{S}, f, order)$, where $\mathcal{S}$ is a connected subset of data points, and f is an element of $\mathcal{F}_{order}$ that approximates every point of $\mathcal{S}$ well. A point x is well approximated by the curve or surface defined by the parameter vector α if the approximate distance satisfies the inequality

$$\delta(\alpha, x)^2 < \epsilon_2 \, \hat{\sigma}_{\mathcal{S}}^2$$

where ϵ_2 is the same constant of the previous paragraph. In the case of surfaces, we also test surface normal continuity. When the noise variance is estimated at a point, the equation of the fitting line or plane also provides an estimate for the curve or surface normal at the point. If the angle between the gradient of the surface defined by α at a point under test and the estimate for the surface normal at the same point is larger than a certain value, the point is rejected.

The region growing starts by finding a *seed region* $\mathcal{R} = (\mathcal{S}, f, 1)$, where $\mathcal{S}$ is a subset of data points, and f is an element of $\mathcal{F}_1$, whose set of zeros $Z(f)$ approximates every point of $\mathcal{S}$ well. In our case, $\mathcal{F}_1$ is the family of first-degree polynomials, and a seed region is the subset of data points in the neighborhood of a point not marked as an outlier, which was used to estimate the noise variance at the point, together with the fitted straight line or plane.

Then, given a current region $\mathcal{R} = (\mathcal{S}, f, order)$, the following loop is repeated until no further growth in $\mathcal{S}$ is observed. The maximal connected region $\mathcal{S}'$ of points well approximated by f and intersecting the initial seed set is computed. If $\mathcal{S}'$ does not have more points than $\mathcal{S}$, then neither $\mathcal{S}$ nor f are changed, and the loop is exited. Otherwise, a new member f' of $\mathcal{F}_{order}$ is fitted to $\mathcal{S}'$, and if it satisfies the goodness of fit test, the region $\mathcal{R}$ is replaced by $\mathcal{R} = (\mathcal{S}', f', order)$ and the loop repeated. If f' does not satisfy the test, the loop is exited. When the loop is exited, if $order$ is equal to the maximum order $order_{\mathrm{MAX}}$, the region growing is finished returning the current region $\mathcal{R} = (\mathcal{S}, f, order)$. Otherwise, a member f' of $\mathcal{F}_{order+1}$ is fitted to $\mathcal{S}$, and if it satisfies the goodness of fit test, f is replaced by f', $order$ is increased to $order + 1$, that is, $\mathcal{R}$ is replaced by $\mathcal{R} = (\mathcal{S}, f', order+1)$, and the loop is traversed once more. If f' does not satisfy the test, the region growing is finished returning the current region $R = (\mathcal{S}, f, order)$.

In our implementation of the algorithm for planar curve segmentation, the second order corresponds to circles, the

third to general second-degree polynomials, the fourth to third-degree polynomials, and so on. In the case of surfaces, the second-order corresponds to spheres and cylinders, the third order to second-degree polynomials, the fourth to third-degree polynomials, and so on. There are two reasons for introducing an extra family between first- and second-degree polynomials. First, we want to locate large and smooth curve or surface patches without curve or surface normal discontinuities or singularities. Since any pair of straight line or planar patches can be well approximated by the set of zeros of a valid second-degree algebraic curve or surface, where the set of zeros of the product of the two linear equations is without the new family, it is equally likely that a region grows across a curve or surface normal discontinuity than along a smooth curved area. The second reason is that slightly curved areas of the data set that are almost well approximated by a straight line are usually best approximated by second-degree curves or surfaces with many branches, such as pairs of lines or planes, hyperbolas or hyperboloids, and ellipses or ellipsoids with very unequal mean axes.

The reason for introducing the merging process is that once a region has grown large enough, very few new points compatible with the current hypothesized curve or surface can be found, and the fit corresponding to the next order is generally rejected due to the same problem that we just encountered for the case of almost flat data sets approximated by second-degree curves or surfaces. The approximate mean square distance is too small. With the merging algorithm, a larger proportion of points is included in the current region, and there exists a possibility of a successful fit of a higher order.

We have restricted the curve or surface types used in the region-growing process to those that can be computed only with the generalized eigenvector fit algorithm without the need of improving them with the reweight heuristic and the iterative Levenson-Marquardt algorithm, saving time in the computation. For planar curves, we use straight lines and circles. For surfaces, the parallel of circles are spheres and cylinders, but since fitting cylinders require the Levenberg-Marquardt algorithm and a strategy to compute an initial estimate, we also moved the spheres and cylinders to the merging phase, leaving a single-order planar region growing.

Figs. 16 and 17 provide a description of our implementation of the region-growing algorithm for planar curves. The corresponding surface algorithms are simplified versions of the planar curve algorithms described above. **TestOrder**$(\mathcal{R})$ is a procedure that implements the goodness of fit test described in the previous section for the region $\mathcal{R} = (\mathcal{S}, f, order)$, and returns one of three values: *Decrease*, *Accept*, or *Increase*. It is clear what the procedure **FitImplicit**$(order, \mathcal{S})$ does. With respect to **GrowRegion**$(\mathcal{R}, \mathcal{R}')$, the procedure **CompatiblePoints**$(\mathcal{R})$ computes a not-necessarily-connected region intersecting the current region by sequentially dilating the current region and checking for compatible points. **IsCompatible** $(x, f, order)$ just checks that the approximate distance from x to the set $Z(f)$ be within the limits imposed by the second part of the goodness of fit test. In the case of surfaces, it also checks the angular deviation between the estimated normal and the gradient of f at the point. Without this check, some surface patches grow along very thin strips of neighboring transversal patches. **LargestComponent** $(p, \mathcal{S})$ computes the largest connected component of the set $\mathcal{S}$, which is adjacent or contains the point p. Finally, **TooFewPoints**$(order, \mathcal{S})$ rejects a region if it has shrunk to much with respect to the given order.

```
procedure GrowRegionFromPoint (p)
  S := SeedSet (p)
  order := Line
  f := FitImplicit (order, S)
  R := (S, f, order)
  CopyRegion (R, R')
  order_MAX := Circle
  LOOP:
  GrowRegion (R, R')
  TEST:
  if order' = Circle or order = Line
    CopyRegion (R', R)
    if order = Circle then
      order := Line
      f := FitImplicit (order, S)
      if TestOrder (R) ≠ Increase then
        CopyRegion (R, R')
        order_MAX := Line
        goto LOOP
    else if order = Line
      if order_MAX = Circle
        order := Circle
        f := FitImplicit (order, S)
        goto LOOP
  return (R')
```

Fig. 16. Region-growing algorithm for planar curves.

```
procedure GrowRegion (R, R')
  do
    Δ1 := CompatiblePoints (R)
    δ1 := |Δ1|
    S := S ∪ Δ1
    f := FitImplicit (order, S)
    Δ2 := CompatiblePoints (R)
    δ2 := |Δ2|
    S'' := S ∪ Δ2
    S := LargestComponent (p, S'')
    if |S| < |S''|/2 then
      if order = Circle then order := Line
      return
    δ3 := |S''| - |S|
    if |S| > |S'| then
      CopyRegion (R, R')
    δ4 := δ1 + δ2 + δ3
    if TooFewPoints (order, S) then
      if order = Circle then order := Line
      return
  while max{|δ1|, |δ2|, |δ3|, |δ4|} > δ_MIN
  return

procedure CompatiblePoints (R)
  B := {x : ||x|| ≤ radius}
  S' := S
  do
    S := S'
    S' := {x ∈ S ⊕ B : IsCompatible (x, f, order)}
  while R' ≠ R
  return (R)
```

Fig. 17. Core of region-growing algorithm for planar curves.

D. Merging

The variable-order merging tries to find *maximal* subsets of the smooth regions or, depending on the application, groups of smooth regions, whrer each of them is represented as a subset of the set of zeros of a single implicit curve or surface. The merging process can be seen as a generalization of region

```
procedure Merging (order, list)
  heap := ∅
  for R' ∈ list
    for R'' ∈ list\{R'}
      node := MergeAndFit (order, R', R'')
      if node ≠ ∅ then
        Insert (node, heap)
  while heap ≠ ∅
    (R, R', R'') := DeleteMin (heap)
    R' := R
    list := list\{R''}
    for R ∈ list\{R'}
      if S ∩ S' ≠ ∅ then
        Delete (R, heap)
    for R'' ∈ list\{R'}
      if S'' ∩ S' ≠ ∅ then
        node := MergeAndFit (order, (R', R'')
        if R ≠ ∅ then
          Insert (node, heap)
  return (list)


procedure MergeAndFit (order, R', R'')
  if order < MinOrder (order', order'') then
    return (∅)
  if order > MaxOrder (order', order'') then
    return (∅)
  S := S' ∪ S''
  f := FitImplicit (order, S)
  if TestOrder (R) ≠ Accept then
    R := ∅
  return ((R, R', R''))
```

Fig. 18. Merging procedure.

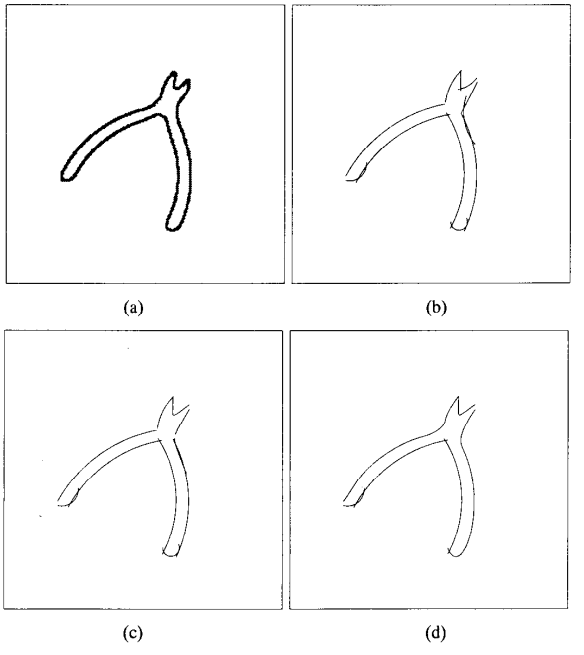


Fig. 19. Segmentation of planar curve: (a) Data set; (b) segmentation after region growing with straight line and circle primitives; (c) after merging (b) using general second-degree algebraic curve primitives (conics); (d) after merging (c) using general third-degree algebraic curve primitives.

growing, where regions grow not by single points but by groups of points.

At every step, the variable-order merging produces the best possible merge of two neighboring regions. At the beginning, all the neighboring pairs that are well fitted by a surface of the current order are computed and inserted into a priority queue according to the value of the maximum relative approximate square distance, which is the number

$$\frac{\delta_{\mathcal{S}}^2(\alpha)}{\Delta_{\mathcal{S}}^2(\alpha)}$$

where $\mathcal{S}$ is the union of the two sets being considered for merging. The priority queue is implemented with a binary heap. Every node of this heap consists of a tern of regions $(\mathcal{R}, \mathcal{R}', \mathcal{R}'')$, where $\mathcal{R}'$ and $\mathcal{R}''$ are two neighboring regions, the set $\mathcal{S}$ of the region $\mathcal{R}$ is the union of the sets $\mathcal{S}'$ and $\mathcal{S}''$ corresponding to the region $\mathcal{R}'$ and $\mathcal{R}'$, and the element f of $\mathcal{R}$ is an acceptable fit of the current *order* for $\mathcal{S}$.

Then, while the queue is not empty, the pair corresponding to the minimum value is deleted from the queue and merged, creating a new region and deleting two, where all the pairs that are still in the queue and involve either one of the two merged regions are deleted, and all the pairs that involve the new region are recomputed and reinserted in the queue. The procedure **Merging**$(order, list)$, which is described in Fig. 18, is called sequentially for increasing values of *order* until the maximum one, where *list* is equal to the current list of regions. The procedure **MergeAndFit**$(order, \mathcal{R}, \mathcal{R}')$ fits a surface of the requested order to the union of the two regions and returns a nonempty data structure only if a fit of the given order successfully passes the goodness of fit test. Finally, the procedures **MinOrder** and **MaxOrder** impose preliminary limits before the fitting precess. For example, if two regions are currently approximated by quadrics, it does not make sense to fit a plane to the union, nor does it make sence to fit an algebraic surface of degree higher than four because the product of the two quadrics already approximates the union well.

XV. Experimental Results

Figs. 19–21 show the results of our segmentation algorithm applied to contours obtained by thresholding gray-level images taken by a standard TV-quality CCD camera. Figs. 22 and 23 show the corresponding results for the segmentation of range images taken with a White Scanner model 100 laser range-finding system, and finally, Fig. 24 shows the segmentation of one range image from the NRCC data base [62] (the file "jet4").

XVI. Conclusions

We show that families of planar curves in x-y, and 3-D surfaces in x-y-z can be described as a subset of a single implicit curve or surface. Families of nonplanar 3-D curves can be described as a subset of the intersection of a pair of 3-D surfaces in x-y-z. We show how this unified representation can be used for position estimation and object recognition.

From an intuitive idea, we developed an approximate distance from a point to a curve or surface defined by implicit equations, turning the problem of fitting curves and surfaces into the minimization of the approximate mean square distance.

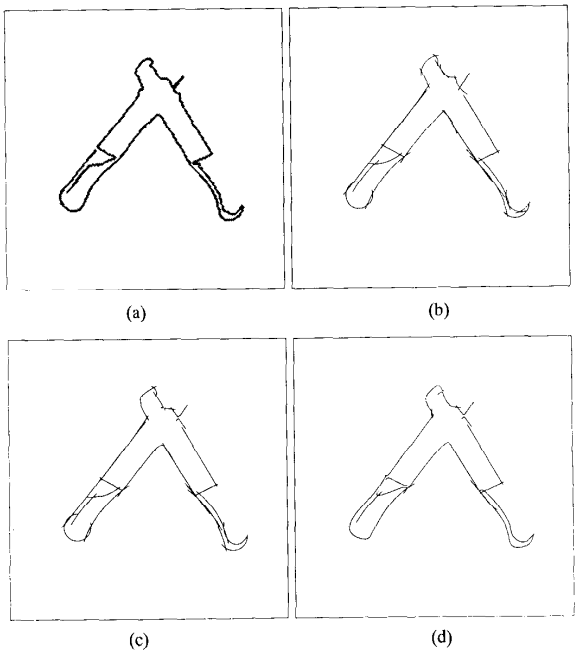


Fig. 20. Segmentation of planar curve: (a) Data set; (b) segmentation after region growing with straight line and circle primitives; (c) after merging (b) using general second-degree algebraic curve primitives (conics); (d) after merging (c) using general fourth-degree algebraic curve primitives.

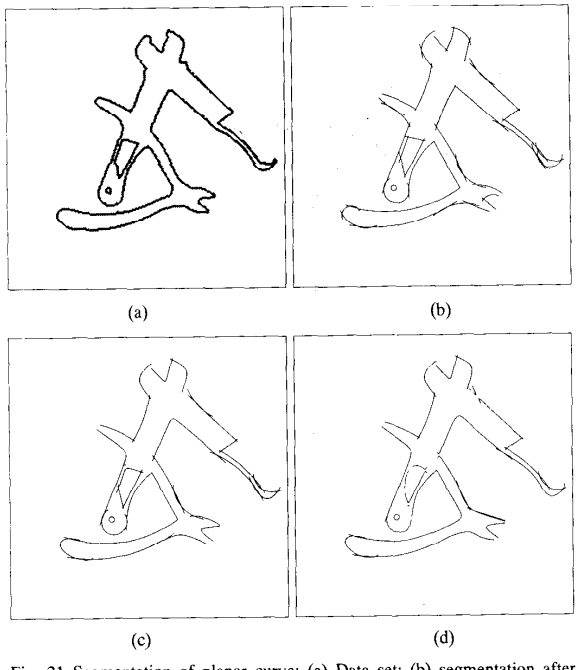


Fig. 21 Segmentation of planar curve: (a) Data set; (b) segmentation after region growing with straight line and circle primitives; (c) after merging (b) using general second-degree algebraic curve primitives (conics); (d); after merging (c) using general third-degree algebraic curve primitives.

We showed that the minimization of the approximate mean square distance reduces to a generalized eigenvector computation for certain families of nonsingular curves and surfaces, and based on this result, we introduced an efficient procedure to compute an initial estimate for the general case.

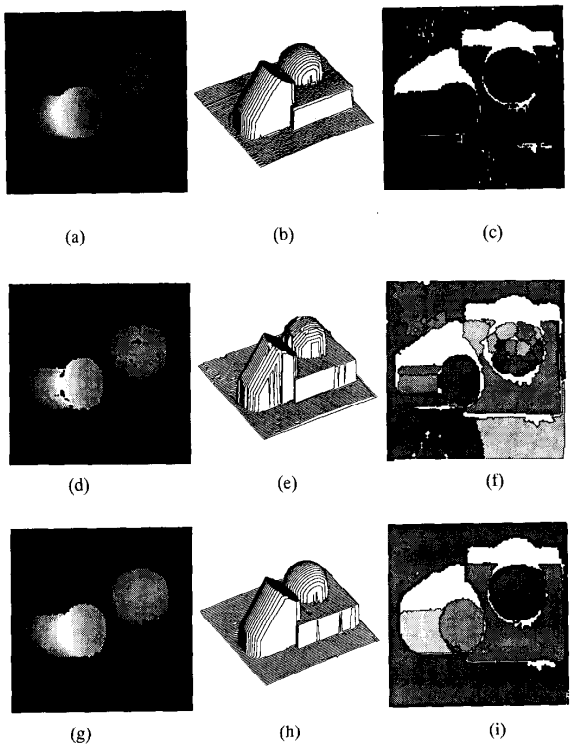


Fig. 22. Segmentation of 3-D scene: sphere, cylinder, and box: (a) Original range image represented as a gray-level image; (b) same range image from a perspective view; (c) white areas represent regions occluded by shadows; (d) result of region growing with planar patches reconstructed as a gray-level image; (e) same result from a perspective view; (f) segmentation after the region growing; (g) result of merging (d)–(f) with general second-degree algebraic surface patches (quadrics), reconstructed as a gray-level image; (h) same result from a perspective view; (i) segmentation after the merging.

The basic contributions of this paper are the approximate distance, the generalized eigenvector fit, and the techniques to minimize the approximate mean square distance.

Since the properties of implicit curves and surfaces are complicated and have received little attention in the computer vision literature, many examples of resulting representations are shown, and computational considerations are discussed.

We show an important set of potential applications to object recognition for the methods introduced in this paper. Among these is the concept of "interest region," which is a generalization of high curvature points.

Finally, we described a variable-order segmentation algorithm for planar curves, surfaces, and space curves, based on the methods introduced in this paper.

APPENDIX A
THE LEVENBERG-MARQUARDT ALGORITHM

Let $\mathcal{D} = \{p_1, \ldots, p_q\}$ be a set of n-dimensional data points, and let $\phi(\alpha, x)$ be a parameterization of the family of

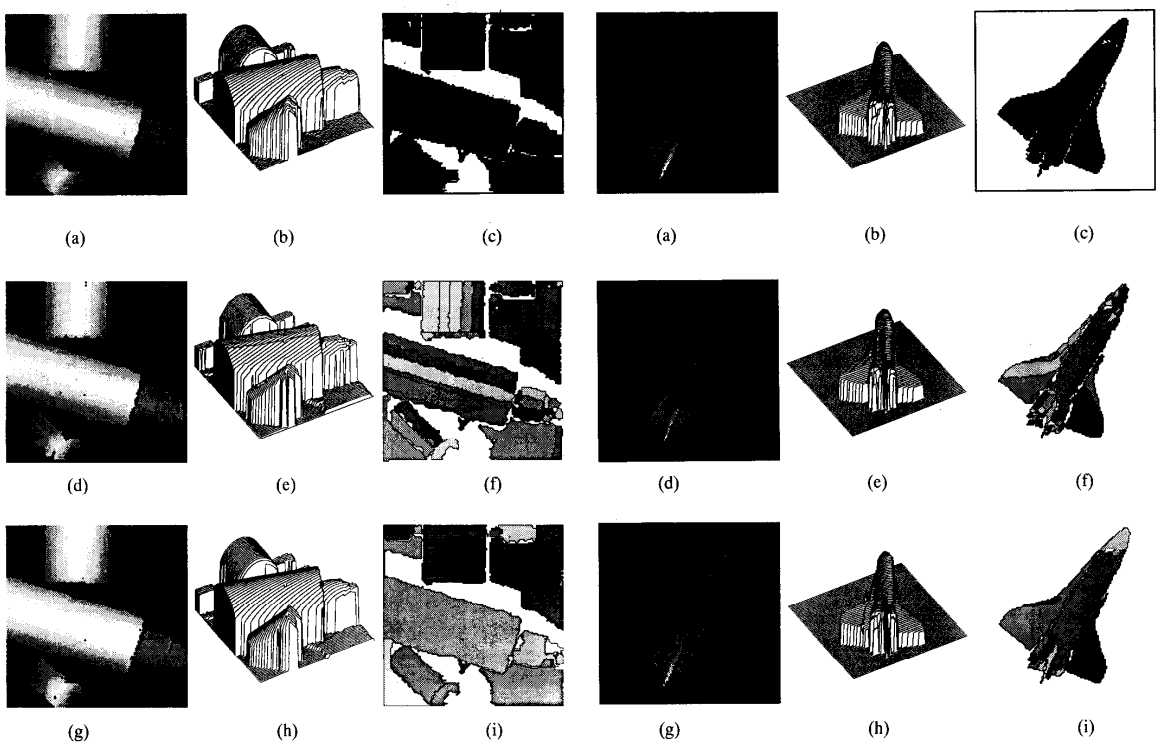


Fig. 23. Segmentation of 3-D scene: spray bottle and cylinders: (a) Original range image represented as a gray-level image; (b) same range image from a perspective view; (c) white areas represent regions occluded by shadows; (d) result of region growing with planar patches reconstructed as a gray-level image; (e) same result from a perspective view; (f) segmentation after the region growing; (g) result of merging (d)–(f) with general second-degree algebraic surface patches (quadrics) reconstructed as a gray-level image; (h) same result from a perspective view; (i) segmentation after the merging.

Fig. 24 Segmentation of 3-D scene: NRCC "jet4" file: (a) Original range image represented as a gray-level image; (b) same range image from a perspective view; (c) white areas represent regions occluded by shadows and the background. (d) result of region growing with planar patches reconstructed as a gray-level image;(e) same result from a perspective view; (f) segmentation after the region growing; (g) result of merging (d)–(f) with general second-degree algebraic surface patches (quadrics), reconstructed as a gray-level image; (h) same result from a perspective view; (i) segmentation after the merging.

admissible maps $\mathcal{F}$. The nonlinear least squares problem is to minimize the length of the *residual vector* $R = (R_1, \ldots, R_q)^t$

$$\|R(\alpha)\|^2 = \sum_{i=1}^{q} R_i(\alpha)^2 = q\, \Delta_D^2(\alpha)$$

where in our case

$$R_i(\alpha) = \delta(\alpha, p_i) \qquad i = 1, \ldots, q$$

and the number of points is not less than the number of parameters. The Levenberg-Marquardt algorithm, which is one of several methods to solve the nonlinear least squares problem, is based on the following iteration step:

$$\alpha^{n+1} = \alpha^n - \left(J(\alpha^n)J(\alpha^n)^t + \mu_n I_q\right)^{-1} J(\alpha^n)^t R(\alpha^n)$$

where $J(\alpha)$ is the Jacobian of R with respect to α

$$J_{ij}(\alpha) = \frac{\partial R_i}{\partial \alpha_j}(\alpha)$$

for $i = 1, \ldots, q \quad j = 1, \ldots, r$, and the constant μ_n is chosen as a small nonnegative number (equal to zero whenever possible), which makes the matrix

$$J(\alpha^n)J(\alpha^n)^t + \mu_n I_q$$

safely positive definite. This strategy assures that the algorithm reduces the value of $\|R(\alpha)\|^2$ at each iteration, converging to a local minimum with fast quadratic local convergence. See Dennis and Shnabel [27] for details.

We only need to provide procedures to compute the values of the residual vector $R(\alpha)$ and the Jacobian $J(\alpha)$. Since all the components of the residual vector have the same form, we only need a procedure to compute $\delta(\alpha, x)$ and its partial derivatives with respect to $\alpha_1, \ldots, \alpha_r$.

For example, let us consider the linear parameterization of planar curves and surfaces $k = 1$. In this case, we have $f(x) = \phi(\alpha, x) = FX$, where $F = \alpha^t$ is an r-dimensional row vector

$$\delta(F, x) = \left(\frac{F[XX^t]F^t}{F[DXDX^t]F^t}\right)^{1/2} = \left(\frac{f(x)^2}{\|\nabla f(x)\|^2}\right)^{1/2}$$

and

$$\frac{\partial \delta}{\partial F_j}(F, x) = \frac{\{[FX]X_j - \delta(F, x)^2[FDX]DX_j^t\}}{\delta(F, x)[FDX][FDX]^t}.$$

In the case of cylinders and, in general, in all the cases of parameterized families of polynomials where we can write $f(x) = FX$ with the coefficient vector as a function of the parameters $F = F(\alpha)$, we just apply the chain rule to the previous expression

$$\frac{\partial \delta}{\partial \alpha_i}(\alpha, x) = \sum_{j=1}^{h} \frac{\partial \delta}{\partial F_j}(F(\alpha), x) \frac{\partial F_j}{\alpha_i}(\alpha).$$

APPENDIX B
Analysis of the Generalized Eigenvector Fit

Let M and N be $r \times r$ symmetric matrices, where N is nonnegative definite, and let F be a $k \times r$ matrix. Let us consider the problem of minimizing

$$\text{trace } (FMF^t) \tag{16}$$

constrained by

$$FNF^t = I_k. \tag{17}$$

Let $h = \text{rank}(N)$. Then, let $h \geq k$; otherwise, the problem has no solution because from (17)

$$h = \text{rank}(N) \geq \text{rank}(FNF^t) = k.$$

We will see that this condition is also sufficient for the existence of a solution. In addition, $k = \text{rank}(F)$ because

$$k = \min\{k, h\} \geq \text{rank}(F) \geq \text{rank}(FNF^t) = k.$$

Let F_i denote the ith row of F and D_{F_i} the Jacobian with respect to the r variables of F_i. We can rewrite (16) as

$$\sum_{i=1}^{k} F_i M F_i^t$$

and (17) as

$$F_i N F_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad i, j = 1, \ldots, k.$$

Let us consider the function

$$\varphi(F, \lambda) = \sum_{l=1}^{k} F_l M F_l^t - \sum_{l=1}^{k} \sum_{j=1}^{k} \lambda_{lj} [F_l N F_j^t - \delta_{lj}]$$

where λ is a $k \times k$ matrix. If $\hat{F}_1, \ldots, \hat{F}_k$ are the rows of a minimizer $\hat{F}$ of (16)–(17), by the Lagrange multipliers theorem (see Theorem 7–10 of [2]), there exists a matrix $\hat{\lambda}$ such that

$$0 = \frac{1}{2} D_{F_i}[\varphi](\hat{F}, \hat{\lambda}) = \hat{F}_i M - \sum_{j=1}^{k} \hat{\lambda}_{ij} \hat{F}_j N$$

for $i = 1, \ldots, k$, or in matrix form

$$\hat{F} M = \hat{\lambda} \hat{F} N.$$

Since the matrix M is symmetric and nonnegative definite, so is FMF^t for every matrix F, and for certain orthogonal $k \times k$ matrix Q

$$Q[FMF^t]Q^t = [QF]M[QF]^t$$

is diagonal. Since $Q^t = Q^{-1}$ and (16) and (17) are invariant with respect to similarity transformations

$$\text{trace } (FMF^t) = \text{trace } ([QF]M[QF]^t)$$

and

$$FNF^t = I_k \qquad \Leftrightarrow \qquad [QF]N[QF]^t = I_k$$

the matrix F is a solution of the minimization problem if and only if QF is. We may assume without loss of generality that $\hat{F}M\hat{F}^t$ is diagonal. In this case, $\hat{\lambda}$ is nonnegative definite and diagonal because

$$\hat{\lambda} = \hat{\lambda}\, I_k = \hat{\lambda}\, \hat{F} N \hat{F}^t = \hat{F} M \hat{F}^t$$

or equivalently

$$\hat{F}_i M = \hat{\lambda}_{ii} \hat{F}_i N \qquad i = 1, \ldots, k$$

$\hat{F}_i$ is a generalized eigenvector of the symmetric-positive pencil $M - \lambda N$ corresponding to the eigenvalue $\hat{\lambda}_{ii}$. It is clear that if $0 \leq \lambda_1 \leq \cdots \leq \lambda_h$ are the eigenvalues of $M - \lambda N$, the choice $\hat{\lambda}_{11} = \lambda_1, \ldots, \hat{\lambda}_{kk} = \lambda_k$ defines a solution of the minimization problem, and a solution clearly exists. The solution is unique if and only if $\lambda_k < \lambda_{k+1}$, and it is a subspace of dimension $s \leq h - k$ if $\lambda_k = \lambda_{k+1} = \cdots = \lambda_{k+s}$ and $s = h - k$ or $\lambda_{k+s} < \lambda_{k+s+1}$.

The pencil $M - \lambda N$ is *symmetric positive* if N is also positive definite. Algorithms to solve a symmetric-positive eigenproblem are well known (see section 8.6. of [42]), and very good subroutine packages are available such as EISPACK [68], [36]. The following is a brief description of such an algorithm.

Since N is symmetric and positive definite, it has a Cholesky decomposition $L^{-1}NL^{-t} = I_r$, where L is nonsingular and lower triangular. Let $0 \leq \lambda_1 \leq \cdots \leq \lambda_k$ be the least eigenvalues of $H = L^{-1}ML^{-t}$, and let $U_1, \ldots, U_k$ be corresponding orthogonal row eigenvectors. The eigenvalues can be computed with the QR algorithm, and then, the eigenvectors can be computed one by one by inverse iteration. If U is the $k \times r$ matrix with rows $U_1, \ldots, U_k$, $\lambda = \text{diag}(\lambda_1, \ldots, \lambda_k)$ and $F = UL^{-1}$, then

$$FM = UHL^t = \lambda U L^t = \lambda F N$$

$$FNF^t = UU^t = I_k$$

and F is the solution of the original problem. F can be computed by backsubstitution.

If N is rank deficient, as it is, for example, in the case of polynomials, we can reduce the original problem to a symmetric-positive one, generalizing the previous algorithm [43].

Since N is nonnegative definite, using the Cholesky decomposition with full pivoting, we can find an $r \times h$ matrix L_1 and an $r \times (r - h)$ matrix L_2 such that $N = L_1 L_1^t$, $NL_2 = 0$, and the square matrix $L = [L_1 L_2]$ is nonsingular. The matrix L is a product of transpositions and elementary triangular transformations. In this case, the matrix $L^{-1}NL^{-t}$ is not the

identity matrix but a diagonal matrix with h ones followed by $r-h$ zeros on the diagonal. The matrix $H = L^{-1}ML^{-t}$ can be decomposed in the following way:

$$H = \begin{pmatrix} H_3 + H_2H_1H_2^t & H_2H_1 \\ H_1H_2^t & H_1 \end{pmatrix} = \begin{pmatrix} I_h & H_2 \\ 0 & I_{(r-h)} \end{pmatrix} \begin{pmatrix} H_3 & 0 \\ 0 & H_1 \end{pmatrix} \begin{pmatrix} I_h & H_2 \\ 0 & I_{(r-h)} \end{pmatrix}^t.$$

The matrices H_1, H_2, H_3 are computed in this order from the corresponding blocks of H. Let U_1 be a $k \times h$ matrix, U_2 a $k \times (r-h)$ matrix, and $U = [U_1U_2]$. Since L is nonsingular, we can write $F = UL^{-1}$ for a certain matrix U. Then,

$$\begin{aligned} &\text{trace } (FMF^t) \\ &\quad = \text{trace } (UHU^t) \\ &\quad = \text{trace } (U_1H_3U_1^t) \\ &\qquad + \text{trace } \left((U_1H_2 + U_2)H_1(U_1H_2 + U_2)^t\right) \end{aligned}$$

and

$$FNF^t = I_k \qquad \Leftrightarrow \qquad U_1U_1^t = I_k.$$

Since U_2 is not included in the constraint and the matrix $(U_1H_2 + U_2)H_1(U_1H_2 + U_2)^t$ is nonnegative definite, if F is a solution of the minimization problem, then $U_2 = -U_1H_2$. The original problem has been reduced to minimizing trace $(U_1H_3U_1^t)$ constrained by $U_1U_1^t = I_k$, which is a particular case of the symmetric positive problem.

ACKNOWLEDGMENT

The author is grateful to Professor D. B. Cooper for all his valuable technical suggestions and comments on previous drafts and his help and advice during the course of this work.

The author would also like to thank the Exploratory Computer Vision Group at the IBM T. J. Watson Research Center, in particular R. M. Bolle, for his support and advise during the development of a good part of this paper, and R. Kjeldsen, for providing the range images used in some of the examples.

REFERENCES

[1] A. Albano, "Representation of digitized contours in terms of conic arcs and straight-line segments," *Comput. Graphics Image Processing*, vol. 3, pp. 23–33, 1974.

[2] T. M. Apostol, *Mathematical Analysis*. Reading, MA: Addison Wesley, 1974.

[3] R. Bajcsy and F. Solina, "Three dimensional object representation revisited," in *Proc. First Int. Conf. Comput Vision*, June 1987, pp. 231–240,

[4] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graphics Applications*, vol. 1, pp. 11–23, 1981.

[5] P. J. Besl, "Geometric modeling and computer vision," *Proc. IEEE*, vol. 76, no. 8, pp. 936–958, 1988.

[6] ——, *Surfaces in Range Image Understanding*. Berlin: Springer-Verlag, 1988.

[7] P. J. Besl and R. C. Jain, "Segmentation trough variable-order surfaces fitting," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, Mar. 1988.

[8] R. H. Biggerstaff, "Three variation in dental arch form estimated by a quadratic equation," *J. Dental Res.*, vol. 51, p. 1509, 1972.

[9] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3D shape by approximating image intensity functions with quadric polynomials," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 4, July 1984.

[10] ——, "On optimally combining pieces of information, with applications to estimating 3D complex-object position from range data," *IEEE Trans. Patt. Analysis Machine Intell.*, vol. 8, no. 5, Sept. 1986.

[11] R. J. Bolle and B. C. Vemuri, "On 3D surface reconstruction methods," RC-14557, IBM Res. Div., Apr. 1989.

[12] R. C. Bolles and P. Horaud, "3DPO: A three-demensinal part orientation system," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 3–26, 1986.

[13] R. C. Bolles, P. Horaud, and M. J. Hannah, "3DPO: A three-dimensinal part orientation sytem," in *Proc. 8th. Int. Joint Conf. Artificial Intell.*, 1983.

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

[15] T. E. Boult and A. D. Gross, "Recovery of superquadrics from depth information," in *Proc. AAAI Workshop Spatial Reasoning Multisensor Integration*, Oct. 1987.

[16] ——, "On the recovery of superellipsoids," CUCS 331–88, Comput. Sci. Dept., Columbia Univ., 1988.

[17] R. A. Bradley and S. S. Srivastava, "Correlation in polynomial regression," *Amer. Statistician*, vol. 33, p. 11, 1979.

[18] J. F. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1988.

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

[20] D. S. Chen, "A data-driven intermediate level feature extraction algorithm," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 7, July 1989.

[21] D. B. Cooper, Y. P. Hung, and G. Taubinl, "A new model-based stereo approach for 3D surface reconstruction using contours on the surfaces pattern, in *Proc. Second Int. Conf. Comput. Vision*, Dec. 1988.

[22] D. B. Cooper and N. Yalabik, "On the cost of approximating and recognizing noise-perturbed straight ines and quadratic curve segments in the plane," NASA-NSG-5035/I, Brown Univ., Mar. 1975.

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

[24] C. P. Cox. *A Handbook of Introductory Statistical Methods*. New York: Wiley, 1987.

[25] P. J. Davis, *Interpolation and Approximation*. New York: Dover, 1975.

[26] C. De Boor, *A Practical Guide to Splines*. Berlin: Springer-Verlag, 1978.

[27] J. E. Dennis and R. B. Shnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[28] A. L. Dixon, "The element of three quantics in two independent variables," in *Proc. London Math. Soc.*, vol. 7, 1908, series 2.

[29] R. O Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[30] I. Elliot, "Discussion and implementation description of an experiemental interactive superquadric-based 3D drawing sytem," Internal Rep. LP 2/IME4, Euro. Comput.-Industry Res. Cent., 1986.

[31] R. T. Farouki and V. T. Rajan, "On the numerical condition of algebraic curves and surfaces," RC-126263, IBM Res. Div., Nov. 1987.

[32] ——, "On the numberical condition of Bernstein ploynomicals," RC-13626, IBM Res. Div., Mar. 1987.

[33] O. D. Faugeras and M. Hebert, "A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. 8th. Int. Joint Conf. Artificial Intell. (IJCAI)*, 1983.

[34] O. D. Faugeras, M. Hebert, and E. Pauchon," Segmentationof range data into planar and quadric patches," in *Proc. IEEE Conf. Comput. Vision Patt. Recog. (CVPR)*, 1983.

[35] Y. Gardan, *Numberical Methods for CAD*. Cambridge, MA: MIT Press, 1986.

[36] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and C. B. Moller, *Matrix Eigensystem Routines—EISPACK Guide Extension*, volume 51 of *Lecture Notes in Computer Science*. Berlin: Springer-Verlag, 1977.

[37] M. Gardiner, "The superellipse: A curve that lies between the ellipse and the rectangle, *Sci. Amer.*, vol. 213, no. 3, pp. 222–234, Sept. 1965.

[38] D. B. Gennery, "Object detection and measurement using stereo vision," in *Proc. DARPA Image Understanding Workshop*, Apr. 1980, pp. 161–167,

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

[40] R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations*. New York: Wiley, 1977.

[41] R. N. Goldman and T. W. Sederberg, "Some applications of resultants to problems in computational geometry," *Visual Comput.*, vol. 1, pp. 101–107, 1985.

[42] G. Golub and C. F. Van Loan, *Matrix Computaitons*. Baltimore: John Hopkins University Press, 1983.

[43] G. H. Golub and R. Underwood, "Stationary values of the ratio of quadratic forms subject to linear constraints," *Z. Angew. Math. Phys.*, vol. 21, pp. 318–326, 1970.

[44] A. D. Gross and T. E. Boult, "Error of fit measures for recovering parametric solids," in *Proc. Second Int. Conf. Comput. Vision*, Dec. 1988.
[45] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision, *IEEE Comput.*, Dec. 1982, pp. 42–54.
[46] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, pp. 417–444, 498–520, 1933.
[47] E. Isaacson and H.B. Keller, *Analysis of Numberical Methods*. New York: Wiley, 1966.
[48] I. T. Jollife, *Principal Component Analysis*. New York: Springer-Verlag, 1986.
[49] K. Levenberg, "A method for the solution of certain problems in least squares, *Quar. Appl. Math.*, vol. 2, pp. 164–168, 1944.
[50] E. A. Lord and C. B. Wilson, *The Mathematical Description of Shape and Form*. Chichester: Ellis Horwood, 1984.
[51] F. S. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge, UK: Cambridge University Press, 1916.
[52] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math*, vol. 11, pp. 431–441, 1963.
[53] J. J. More, B. S. Garbow, and K.E. Hillstrom, "User guide for minpack-1," ANL-080–74, Argone Nat. Lab., 1980.
[54] K. A. Paton, "Conic section in automatic chromosome analysis," *Machine Intell.,*, vol. 5, p. 411, 1970.
[55] ——, "Conic sections in chromosome analysis," *Patt. Recogn.*, vol. 2, p. 39, 1970.
[56] C. Pearson, *Numerical Methods in Engineering and Science*. New York: Van Nostrand Reinhold, 1986.
[57] K. Pearson, "On lines and planes of closest fit to systems fo points in space," *Philos. Mag.*, vol. 2, p. 559, 1901, series 2.
[58] A. P. Pentland, "Recognition by parts," in *Proc. First Int. Conf. Comput. Vision*, June 1987, pp. 612–620.
[59] J. Ponce and D. J. Kriegman, "On recognizing and positioning curved 3D objects from image contours," in *Proc. IEEE Workskhop Interpretation 3D Scenes*, Nov. 1989.
[60] V. Pratt, "Direct least squares fititng of algebraic surfaces, *Comput. Graphics*, vol. 21, no. 4, pp. 145–152, July 1987.
[61] J. O. Ramsay, "A comparative study of sevseral robust estimates of slope, intercept, and scale in linear regression," *J Amer. Stat. Assoc.*, vol. 72, pp. 608–615, 1977.
[62] M. Rioux and L. Cournoyer, "The NRCC three-dimensional image data files," CNRC 29077, Nat. Res. Council Canada, June 1988.
[63] G. Salmon, *Modern Higher Algebra*. Dublin: Hodges, Smith and Co., 1866.
[64] P. D. Sampson, "Fitting conic sections to very scattered data: An iterative refinement of the Bookstein algorithm, *Comput. Vision Graphics Image Processing*, vol. 18, pp. 97–108, 1982.
[65] J. Schwartz and M. Sharir, "Identificatin of partially obscured objects in two and three dimensions by matching noisy characteristic curves," *Int. J. Robotics Res.*, vol. 6, no. 2, 1987.
[66] T. W. Sederberg, D. C. Anderson, and R. N. Goldman, "Implicit representation of parametric curves and surfaces," *Comput. Vision Graphics Image Processing*, vol. 28, pp. 72–84, 1984.
[67] J. F. Silverman and D. B. Cooper, "Bayesian clustering for unsupervised estimation of surface and texture models," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, July 1988.
[68] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moller. *Matrix Eigensystem Routines-EISPACK Guide*, volume 6 of *Lecture Notes in Computer Science*. New York: Springer-Verlag, 1976.
[69] F. Solina, "Shape recovery and segmentation with deformable part models," Ph.D. thesis, Univ. Pennsylvania, Dec. 1987.
[70] G. Taubin, "Algebraic nonplanar curve and surface estimation in 3–space with applications to positoin estimation," Tech. Rep. LEMS-43, Brown Univ., Feb. 1988.
[71] ——, "Nonplanar curve and surface estimation in 3–space," in *Proc. IEEE Conf. Robotics Automation*, May 1988.
[72] K. Turner, "Computer perception of curved objects using a television camera," Ph.D. thesis, Univ. Edinburgh, Nov. 1974.
[73] B. C. Vemuri, "Representation and recognition of objects from dense range maps," Ph.D. thesis, Univ. Texas at Austin, Dept. Elect. Eng., 1987.
[74] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representations of objects from range data," *Image Vision Comput.*, vol. 4, no. 2, pp. 107–114, May 1986.
[75] R. Walker, *Algebraic Curves*. Princeton, NJ: Princeton University Press, 1950.
[76] H. Weyl, *The Classical Groups*. Princeton, NJ: Princeton University Press, 1939.

Gabriel Taubin (M'90) was born on June 12, 1958 in Morón Province of Buenos Aires, Argentina. He received the "Licenciado en Ciencias Matematicas" degree from the University of Buenos Aires, Argentina, in 1981 and the Ph.D. degree in electrical engineering from Brown University, Providence, RI, in 1991.

During the period of four years of his residence at Brown University, he spent two summers at the IBM T.J. Watson Research Center. He was recipient of a Brown University fellowship and two IBM Fellowships. Currently, his research interests are in the applications of mathematics, particularly geometry, to computer vision and robotics problems and more generally to the broad range of problems in intelligent machines.

Mr. Taubin is a member of the Society of Industrial and Applied Mathematics (SIAM).