Implicit Polynomial Representation Through a Fast Fitting Error Estimation

Mohammad Rouhani, *Student Member, IEEE*, and Angel Domingo Sappa, *Member, IEEE*

***Abstract*—This paper presents a simple distance estimation for implicit polynomial fitting. It is computed as the height of a simplex built between the point and the surface (i.e., a triangle in 2-D or a tetrahedron in 3-D), which is used as a coarse but reliable estimation of the orthogonal distance. The proposed distance can be described as a function of the coefficients of the implicit polynomial. Moreover, it is differentiable and has a smooth behavior. Hence, it can be used in any gradient-based optimization. In this paper, its use in a Levenberg–Marquardt framework is shown, which is particularly devoted for nonlinear least squares problems. The proposed estimation is a generalization of the gradient-based distance estimation, which is widely used in the literature. Experimental results, both in 2-D and 3-D data sets, are provided. Comparisons with state-of-the-art techniques are presented, showing the advantages of the proposed approach.**

***Index Terms*—Curve/surface fitting, geometric distance estimation, implicit polynomial (IP), residual error minimization.**

I. Introduction

IMPLICIT polynomials (IPs) have been used in the computer vision field because they are advantageous compared with other representations. First, they are a compact way to represent a given data set, i.e., in a 2-D/3-D space; second, since they do not require any parametrization, they can be obtained without a prior knowledge about the data point spatial distribution or local neighborhood relationship. They are very attractive particularly when compared with other kinds of data representations that need to know the spatial data distribution (e.g., triangular meshes [1], [2], B-spline, or parametric active contours [3], [4]). IP compactness has been also an attractive point to be exploited when a high-level reasoning is needed (e.g., object recognition [5], object modeling [6], [7], reverse engineering, etc.).

In general, IP representations are obtained through a fitting process. Two different approaches have been proposed in the literature to find the "best" IP fitting the given data set, i.e., 1) algebraic and 2) geometric; their difference depends on the criterion used to define "best" (i.e., accuracy versus speed). The next section briefly details these two approaches.

Manuscript received July 23, 2010; revised December 06, 2010, March 21, 2011, and May 11, 2011; accepted September 12, 2011. Date of publication September 29, 2011; date of current version March 21, 2012. This work was supported in part by the Government of Spain under Research Program Consolider Ingenio 2010: Multimodal Interaction in Pattern Recognition and Computer Vision (CSD2007-00018) and in part by the Ministerio de Ciencia e Innovación under Project TIN2011-25606 and Project TIN2011-29494-C03-02. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sharath Pankanti.

The authors are with the Computer Vision Center, Universitat Autònoma de Barcelona Campus, 08193 Bellaterra, Barcelona, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2170080

This paper has two main contributions. The first contribution is the estimation of the orthogonal distance (Euclidean) through a simple approach, which has been initially proposed for the quadratic IP case [8], [9]. The advantage of the proposed estimation is twofold. First, it provides a more accurate value than current approaches. Second, it can be efficiently computed and run in real time. The second contribution is based on the use of such an estimation in a nonlinear minimization framework, i.e., the Levenberg–Marquardt algorithm (LMA). The rest of this paper is organized as follows. Section II describes the problem and introduces related work. The proposed technique is presented in Section III. Section IV gives the experimental results and comparisons. Finally, conclusions are presented in Section V.

II. Problem Formulation and Related Work

The two major approaches in implicit polynomial fitting, namely, *algebraic* and *geometric*, are briefly presented here to show the motivations of the proposed approach. IP fitting aims at finding the best polynomial that describes a given set of points by means of its *zero set*. In other words, the value of the polynomial should reach zero at the location of the given data points. Let $f_{\mathbf{c}}(\mathbf{x})$ be an implicit polynomial of degree d represented as

$$f_{\mathbf{c}}(\mathbf{x}) = \sum_{\substack{(i+j+k)\leqq d \\ \{i,j,k\}\geqq 0}} c_{i,j,k} \cdot x^i \cdot y^j \cdot z^k \tag{1}$$

or in a vector form

$$f_{\mathbf{c}}(\mathbf{x}) = \mathbf{m}^T\mathbf{c} \tag{2}$$

where $\mathbf{c} = [c_{0,0,0}, c_{1,0,0}, \ldots c_{0,0,d}]^T$ is the column vector of polynomial coefficients having as many components as the combination of $(d+3)$ taken three at a time without repetitions, i.e., $C_3^{d+3} = (d+3)!/d!3!$; and $\mathbf{m}$ is the column vector of monomials, i.e., $\mathbf{m} = \mathbf{m}(\mathbf{x}) = [x^0y^0z^0, x^1y^0z^0, \ldots x^0y^0z^d]^T$; the fitting problem consists of first defining a criterion, or a residual error, for measuring the closeness of zero set $Z_f = \{\mathbf{x} : f_{\mathbf{c}}(\mathbf{x}) = 0\}$ to the given data set, and then minimizing this criterion to find the best coefficient vector $\mathbf{c}$.

Let $P = \Gamma_0 = \{p_i\}_1^N$ be the set of given data points with coordinates $\mathbf{x}$ (picked up from object boundaries in 2-D or surfaces in 3-D); then, the fitting problem is defined as

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \mathrm{Dist}(P, f_{\mathbf{c}}) \tag{3}$$

where $\arg\min_{\mathbf{c}}$ stands for polynomial coefficient vector $\mathbf{c}$, where the Dist expression attains its minimum value; there are two different approaches to find that best coefficient vector $\hat{\mathbf{c}}$ as detailed next.

A. Algebraic Approaches

Since the implicit representation is used, a point is on the surface if and only if the output of $f_{\mathbf{c}}$ in (2) is zero at the given point. It leads us to define the following optimization criterion, which is known as algebraic distance:

$$\mathrm{Dist}(P, f_{\mathbf{c}}) = \sum_{i=1}^{N} f_{\mathbf{c}}^2(p_i). \tag{4}$$

This optimization problem has the trivial answer $\mathbf{c} = \mathbf{0}$, giving zero as a minimal value. In order to avoid the trivial answer, a normalization constraint must be imposed. For example, the two classical normalization constraints used in the literature are 1) forcing the optimum vector to have a unit l_1 length (i.e., $\sum c_i = 1$) or 2) having a unit constant coefficient (i.e., $c_0 = 1$). More elaborated constraints have been also proposed; for instance, [10] imposes the mean value of gradient length to be unit (i.e., $(1/N)\sum_{i=1}^{N} \|\nabla f_{\mathbf{c}}^2(p_i)\|^2 = 1$). Due to simplicity, the second normalization constraint is used in this paper, and the constant element in monomial, together with its corresponding coefficient, is removed in this case. This minimization problem is also equivalent to the overdetermined system of equations

$$\boldsymbol{M}\mathbf{c} = \mathbf{b} \tag{5}$$

where $\boldsymbol{M}$ is the monomial matrix (every row contains monomial vector $\mathbf{m}^T(p_i)$ computed at the given point), and $\mathbf{b} = -\mathbf{1}$ is a column vector containing -1 in every entry. Regardless to these interpretations, the optimal solution could be algebraically computed through least squares solutions

$$\mathbf{c} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}M^T\mathbf{b}. \tag{6}$$

The noniterative framework of algebraic approaches is an appealing feature for many applications. In spite of that, two common problems inherent to algebraic approaches are: 1) computational instability of the zero set and 2) lack of geometric information of the data in this procedure. For instance, focusing on the instability problem, Helzer *et al.* [11] analyzed the sensitivity of the zero set to small coefficient changes and minimized an upper bound of the error in order to have a more stable output. Keren and Gotsman [12] tried to constrain the surface parameter space in order to obtain a geometrically reasonable output. Tasdizen *et al.* [13] proposed adding some geometric concept inside the optimization problem. They try to maintain the estimated gradient value at each data points while they fit the data.

The 3L algorithm proposed by Blane *et al.* [14] is a linear least squares polynomial fitting that consists of generating two additional *level sets*, namely, $\Gamma_{-\delta}$ and $\Gamma_{+\delta}$, from given data set Γ_0. These two additional data sets are generated so that one is internal and the other is external, and they are placed at a distance $\pm\delta$ from the original data along a direction that is locally perpendicular to the given data set. Hence, the 3L algorithm incorporates local geometric information resulting in a more stable solution. Considering the three level sets, i.e., $\{\Gamma_{-\delta}, \Gamma_0, \text{and } \Gamma_{+\delta}\}$, (5) could be represented by using a block matrix $\mathbf{M}_{3L}$ and a block column vector $\mathbf{b}$

$$\mathbf{M}_{3L} = \begin{bmatrix} \mathbf{M}_{\Gamma_{-\delta}} \\ \mathbf{M}_{\Gamma_0} \\ \mathbf{M}_{\Gamma_{+\delta}} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -\boldsymbol{\epsilon} \\ \mathbf{0} \\ +\boldsymbol{\epsilon} \end{bmatrix} \tag{7}$$

where $\mathbf{M}_{\Gamma_0}$, $\mathbf{M}_{\Gamma_{+\delta}}$, and $\mathbf{M}_{\Gamma_{-\delta}}$ are the matrices of monomials calculated in the original, inner, and outer sets, respectively; and $\pm\epsilon$ are the corresponding expected values in the inner and outer level sets. The distance metric proposed by the 3L algorithm is

$$\mathrm{Dist}(P, f_{\mathbf{c}}) = \sum_{\mathbf{x}\in\Gamma_0} f_{\mathbf{c}}(\mathbf{x})^2 + \sum_{\mathbf{x}\in\Gamma_{+\delta}} \left(f_{\mathbf{c}}(\mathbf{x}) - \epsilon\right)^2 + \sum_{\mathbf{x}\in\Gamma_{-\delta}} \left(f_{\mathbf{c}}(\mathbf{x}) + \epsilon\right)^2. \tag{8}$$

Then, the least squares solution for $\mathbf{c}$ is obtained as

$$\mathbf{c} = \mathbf{M}_{3L}^{\dagger}\mathbf{b} = \left(\mathbf{M}_{3L}^T\mathbf{M}_{3L}\right)^{-1}\mathbf{M}_{3L}^T\mathbf{b} \tag{9}$$

where $\mathbf{M}_{3L}^{\dagger}$ denotes the pseudoinverse of $\mathbf{M}_{3L}$. Aiming at improving the accuracy of the 3L algorithm, [15] proposes an algorithm, still in the algebraic category, which relaxes the additional constraints (7) so that the $\pm\epsilon$ values are independently adjusted for every single point.

B. Geometric Approaches

In this case, the distance between a point and the surface is usually defined as the shortest distance between this point and its correspondence on the surface (i.e., orthogonal distance). Thus, in the general case of geometric methods, we have the following optimization problem:

$$\mathrm{Dist}(P, f_{\mathbf{c}}) = \sum_{i=1}^{N} \min_{\hat{p}_i} d^2(p_i, \hat{p}_i) \tag{10}$$

where each $\hat{p}_i$ is the correspondence of p_i on the surface. Here, we consider the l_2 norm to calculate distance d, and consequently, a nonlinear least squares optimization must be solved.

Theoretically, both unknown surface parameters and the correspondences must be simultaneously found, but practically, this problem is tackled by first assuming an initial surface, and then refining it until convergence is reached. Therefore, the fitting problem is split up into two stages, i.e., 1) point correspondence search and 2) surface parameter refinement. The first stage deals with the summands in (10), whereas the second one concerns about (3).

Point Correspondence Search: Regarding the first stage, two different strategies have been proposed in the literature, i.e., 1) finding the shortest distance by solving a nonlinear system (e.g., [16] and [17]) and 2) computing an estimation of the shortest distance (e.g., [10], [18], and [19]).

In [16], Ahn *et al.* proposed a method to find the correspondence (or *foot point*) on the surface, which is based on its geometric properties. This foot point $\hat{p}$ is somewhere on the surface satisfying $f_{\mathbf{c}}(\hat{p}) = 0$. Furthermore, the line connecting the data point with the foot point must be parallel to the $\nabla f_{\mathbf{c}}$ at the foot

point, where ∇ is the gradient operator. In other words, we must have $\nabla f_{\mathbf{c}} \times (\hat{p} - p) = 0$. Merging these two conditions, the following system of equations must be solved:

$$\begin{pmatrix} f_{\mathbf{c}} \\ \nabla f_{\mathbf{c}} \times (\hat{p} - p) \end{pmatrix} = \mathbf{0}. \tag{11}$$

This equation could be solved by the Newton–Rophson algorithm for nonlinear system of equations. Although the fitting method in [16] is precise enough, and even covers some well-known methods in the literature such as [10] and [20], it is quite time consuming due to the iterations.

In [17], the orthogonal fitting is extended for general error functions such as the l_1 and l_∞ norms of the residual error instead of the common l_2 norm. This highlights the importance of the error function selection for the fitting process. The authors present the fitting algorithm as an evolutionary process of a surface along its normal direction. They discuss and compare their approach with other common error functions, including the algebraic types.

Instead of computing the shortest distance through (11), [19] proposes approximating it, avoiding iterative approaches as a result. In that work, which is an extension of [18] for more general surfaces, first, a normal vector $\overrightarrow{n}_p$ for each point p is computed by using the principal component analysis in a small $k \times k$ neighborhood centered at each point [21]. In other words, $\overrightarrow{n}_p = (n_1, n_2, n_3)$ is defined as the eigenvector of local covariance matrix Cov associated with the smallest eigenvalue

$$Cov = \frac{1}{s} \sum_{i=1}^{s} (\mathbf{p}_i - \tilde{\mathbf{p}})(\mathbf{p}_i - \tilde{\mathbf{p}})^T \tag{12}$$

where $\tilde{\mathbf{p}} = (1/s) \sum_{i=1}^{s} p_i$ is the vector showing the mean position of the neighboring points in the $k \times k$ region. Finally, $\hat{p}$ is computed as the intersection of the surface $f_{\mathbf{c}}(\mathbf{x}) = 0$ with a line passing through p and parallel to $\overrightarrow{n}_p$, i.e.,

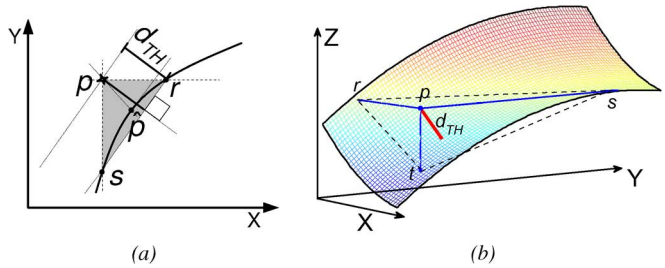
$$\frac{x - x_p}{n_1} = \frac{y - y_p}{n_2} = \frac{z - z_p}{n_3}. \tag{13}$$

The intersection is used as an approximation for the foot point $\hat{p}$ in the geometric distance (10).

In [10], Taubin proposes an approximation for (10), which is based on the first-order Taylor expansion of the distance function. The distance could be computed through normalizing the algebraic distance by the gradient norm

$$\mathrm{Dist}(P, f_{\mathbf{c}}) = \sum_{i=1}^{N} \left(\frac{|f_{\mathbf{c}}(p_i)|}{\|\nabla f_{\mathbf{c}}(p_i)\|} \right)^2. \tag{14}$$

This approximated distance is used in an iterative weighted least squares method and in a nonlinear optimization framework. In addition, a new constraint is imposed on the coefficient vector, which is based on the data points and on the coefficients. The approximated distance proposed by Taubin [10] may not reach the correspondence point lying on the zero set, which could affect the final fitting result. In fact, instead of considering the zero set, the level set where the point is lying on is affected by this optimization process. Finally, every point forces its level set to move in order to reach a lower accumulated distance.



(a) (b)

Fig. 1. Simplex used for estimating the geometric distance: (a) 2-D case and (b) 3-D case.

Surface Parameter Refinement: As a result from the previous stage, the set of points $\{\hat{p}_i\}_{i=1}^{n}$ corresponding to every p_i in the given data set has been found. Afterward, it must be followed by an optimization framework to refine the surface parameter. Although different optimization algorithms could be used (e.g., genetic algorithm (GA) [19], trust region [22], quasi-Newton method [23], and particle swarm [24]), in this paper, the LMA [25] has been chosen since it exploits gradient information provided by the proposed distance estimation. LMA, in some sense, interpolates between the Gauss–Newton algorithm and the gradient descent (more details about the LMA are given in Section III-B).

III. Proposed Approach

This paper proposes a geometric approach to tackle IPs fitting through an estimation of the orthogonal distance. In spite of being focused on the geometric framework, the polynomial coefficients are first initialized by using an algebraic-based algorithm like the 3L algorithm [14]. This initialization process is intended for speeding up the convergence of the algorithm; other strategies, for instance, starting with the smallest bounding circle/sphere, can be used as well. The proposed geometric approach consists of two stages. First, the residual error from the given set of points to the initial IP coefficients is estimated by means of the proposed approach. Then, the IP coefficients are accordingly updated through LMA. The two stages are repeated until convergence is reached; they are detailed next.

A. Approximated Residual Error

The first contribution of this paper lies in a direct approach to estimate the orthogonal distance, which works as follows. First, a *simplex* is constructed through each point and its intersections along the coordinate axis. A simplex is a triangle in 2-D and a tetrahedron in 3-D, as depicted in Fig. 1(a) and (b), respectively. Without loss of generality, the 3-D case is considered here. In this case, having constructed the tetrahedron, its height segment is considered as an approximation of the geometric distance. This tetrahedron is defined by the given point and three intersections satisfying $f_{\mathbf{c}}(x, y_i, z_i) = 0$, $f_{\mathbf{c}}(x_i, y, z_i) = 0$, and $f_{\mathbf{c}}(x_i, y_i, z) = 0$, where $p = p_i(x_i, y_i, z_i)$ is the given point. In the particular case tackled in this paper, since the fitted curve/surface is defined by the implicit polynomial (1), the intersecting points are found by computing the closest root of three 1-D functions to the data point.

Once the intersecting points have been obtained, a direct formula is used to estimate the geometric distance. Let r, s, and t

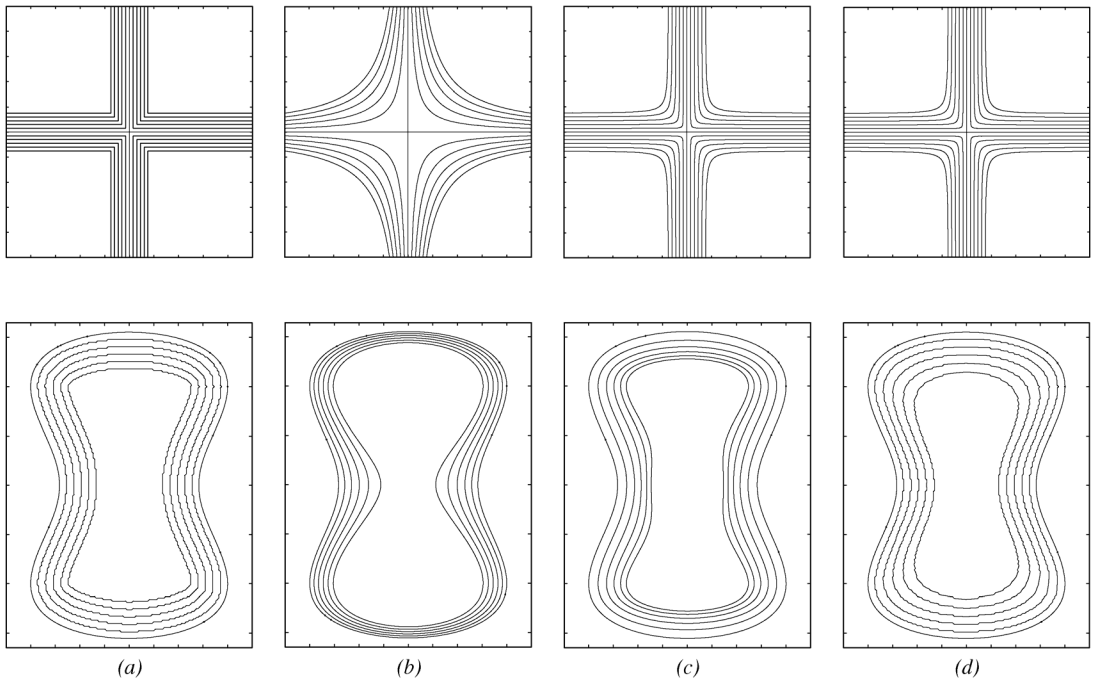


Fig. 2. Contour of constant distance for (a) orthogonal distance; (b) algebraic distance; (c) [10]; and (d) proposed distance estimation.

be the three intersections with the current surface that create a triangular planar patch [see Fig. 1(b)]. Since the volume of the tetrahedron is defined as the product of the area of each base by its corresponding height, three sets of expressions lead us to the same value. Hence, the height of the tetrahedron d_{TH} could be easily computed from the following relationship:

$$\begin{aligned} d_{\mathrm{TH}} &= \left(|pr| \cdot |ps| \cdot |pt|\right) / |\vec{rs} \times \vec{rt}| \\ &= \frac{|pr| \cdot |ps| \cdot |pt|}{\sqrt{(|pr| \cdot |ps|)^2 + (|pr| \cdot |pt|)^2 + (|ps| \cdot |pt|)^2}} \end{aligned} \quad (15)$$

where $\times$ refers to the cross-product operator between two vectors. Similar relationship can be obtained in the 2-D case by using the triangle area instead of the tetrahedron volume. More details can be found in [9].

As presented above, in order to estimate the distance, the intersections of the curve/surface along the coordinate axis must be found first. In the quadric case, these intersections can be directly found [9]. However, for higher degree cases, an iterative method should be used to find the roots. In the current implementation, Newton's method has been used [26]. In the case that the first iteration is considered, an approximation of the root can be obtained through the first-order Taylor expansion. For instance, the expansion along the x axis can be expressed as follows:

$$f(x, y_i, z_i) \simeq f(x_i, y_i, z_i) + f_x(x_i, y_i, z_i) \cdot (x - x_i) \quad (16)$$

where f_x corresponds to the partial derivative in the x-direction, and $x = r$ is the intersection of the surface with the line passing through p in the x-direction. Hence, segment $|pr|$ can be easily estimated as

$$|pr| \simeq -f(p_i)/f_x(p_i). \quad (17)$$

Considering similar approximations for the other two intersections, the proposed distance for point p_i could be approximated as follows:

$$\begin{aligned} d_{\mathrm{TH}} &\simeq \frac{|f/f_x| \cdot |f/f_y| \cdot |f/f_z|}{f^2 \sqrt{(1/f_x \cdot f_y)^2 + (1/f_x \cdot f_z)^2 + (1/f_y \cdot f_z)^2}} \\ &= \frac{|f|}{\|\nabla f\|} \end{aligned} \quad (18)$$

thus, the proposed distance is *a generalization* of Taubin's method when the intersections are approximated.

The preciseness of the proposed distance is presented for two examples in Fig. 2 and compared with other approximated distances, as well as with the orthogonal one. The first row of the figure shows the isocontours[1] of the set $\{(x, y) : xy = 0\}$, which consists of two intersecting lines, and the second row shows the isocontours of a regular curve $\{(x, y) : 8x^2 + (y^2 - 4)^2 - 32 = 0\}$. As illustrated in the last two columns, our method and Taubin's similarly behave in the linear case (when the Jacobian matrix is linear with respect to the point coordinates). In the second example, our method outperforms compared with other approximations and has a quite similar result to the isocontours obtained by the orthogonal method.

B. Implicit Polynomial Fitting

As a result from the previous stage, the distance from each single data point to the current curve/surface has been found. Accumulation of all these distances provides a good criterion for curve surface fitting

$$\mathrm{Dist}(P, f_{\mathbf{c}}) = \sum_{i=1}^{N} d_{\mathrm{TH}}^2(p_i). \quad (19)$$

[1]Contours with the same distance from the zero set.

This distance is in the least squares form where each term is nonlinear with respect to coefficient vector $\mathbf{c}$. It provides a straightforward method to approximate the orthogonal distance. Hence, it can be used in an appropriate optimization algorithm to find the best parameters describing the given set of points. We already used this distance in a RANdom SAmple Consensus-based framework to find the quadratic surface parameters [9]. Other optimization techniques such as GA [19] or quasi-Newton method [23] have been already used in surface fitting.

This paper not only proposes a simple and fast distance estimation but also, as a second contribution, it shows how this estimation can be used in a nonlinear framework. In our implementation, the LMA has been used [25] to optimize the distance (19) with respect to the curve/surface parameters. LMA is specifically designed for nonlinear functions in the least squares form, which is the case in (19). It starts from an initial coefficient vector $\mathbf{c}^0 = \mathbf{c}$, obtained by some algebraic fitting technique (as aforementioned, the result from the 3L algorithm has been used as initialization). LMA updates these parameters iteratively as follows:

$$\begin{aligned} \mathbf{c}^{t+1} &= \mathbf{c}^t + \beta \triangle \mathbf{c} \\ \left(\boldsymbol{J}^T \boldsymbol{J} + \lambda \text{diag}(\boldsymbol{J}^T \boldsymbol{J}) \right) \triangle \mathbf{c} &= J^T \mathbf{D} \end{aligned} \tag{20}$$

where β is the refinement step; $\triangle \mathbf{c}$ represents the refinement vector for the surface parameters; λ is the damping parameter in LMA; $\boldsymbol{J}$ is the Jacobian matrix of $\mathbf{D}$; and vector $\mathbf{D} = (d_1(\mathbf{c}), \ldots, d_n(\mathbf{c}))^T$ corresponds to distances $(d_i(\mathbf{c}) = d_{\text{TH}}(p_i))$, whose l_2 norm must be minimized. Parameter refinement (20) must be repeated until convergence happens.

Each iteration of LMA contains two stages, namely, 1) distance estimation and 2) Jacobian matrix computation. In the first stage, all the intersections along the coordinate axis must be found. For this purpose, Newton's method is applied to find the root of the parametric function $f(\mathbf{p_i} + t\mathbf{d})$, which is a 1-D function with respect to t. Direction vector $\mathbf{d}$ is set to $\boldsymbol{e}_1 = (1,0,0)^T$, $\boldsymbol{e}_2 = (0,1,0)^T$, and $\boldsymbol{e}_3 = (0,0,1)^T$ for each axis. Having computed all the intersections along the coordinate axis, the terms $|pr|$, $|ps|$, and $|pt|$, and consequently the distance (15), can be estimated. As aforementioned, it should be noticed here that if we stop the Newton's method after one iteration, the proposed distance will be easily computed through (18), which is the same as [10].

In order to handle LMA, the value of the functional (19) and its partial derivatives, which are used to build the Jacobian matrix, should be provided. These values show the sensitivity of each d_i in (15) with respect to parameter vector $\mathbf{c}$. The Jacobian matrix could be directly derived through the differentiation rules as follows:

$$\begin{aligned} J_{ij} = \partial_j(d_i) = &\left[|\vec{rs} \times \vec{rt}| \cdot \partial_j \left(|pr| \cdot |ps| \cdot |pt| \right) \right. \\ &\left. - \left(|pr| \cdot |ps| \cdot |pt| \right) \cdot \partial_j |\vec{rs} \times \vec{rt}| \right] / |\vec{rs} \times \vec{rt}|^2 \end{aligned} \tag{21}$$

where $\partial_j = \partial / \partial c_j$ is the differentiation operator with respect to parameters. Since the intersection r, s, and t lies on the surface, $|\vec{rs} \times \vec{rt}|$, $|pr|$, $|ps|$, and $|pt|$ can be implicitly expressed as a function of the surface parameters. In order to calculate each term of (21), the implicit differentiation rule must be used for each intersection. For instance, for a given point p_i, term $|pr|$ is computed just by considering its x-component, i.e., $|pr| = (r_i^x - p_i^x)$ and its partial derivatives as follows:

$$\partial_j |pr| = \frac{dr^x}{dc_j} = -\frac{\partial f / \partial c_j}{\partial f / \partial r^x} = -\frac{m_j(r)}{f_x(r)} \tag{22}$$

where $m_j(r)$ is the jth monomial component calculated in the intersection. Term $|\vec{rs} \times \vec{rt}|$ can be expressed based on the intersections, as mentioned in (15). Then, its partial derivatives can be computed based on the other single terms.

Having estimated the geometric distance (15) and its Jacobian matrix through (21), the LMA iterates equations in (20) until convergence is reached. In this paper, a convergence criterion has been defined using the deviation between the IP normal and the local normal at each point (see illustration in Fig. 3). This criterion, on the one hand, is easy to be computed, and on the other hand, it is robust enough to be used with different geometries. Note that the local normal at each data is already computed during the initialization stage (the 3L algorithm). Therefore, the only required computation is regarding the angle estimation

$$\theta_i = \cos^{-1} \left(\frac{\boldsymbol{n}_i \cdot \nabla f(p_i)}{\| \nabla f(p_i) \|} \right). \tag{23}$$

Additionally, since $\cos^{-1}|_{[0,1] \to [0, pi/2]}$ is monotonic, just the absolute value of the inner expression, without calculating the

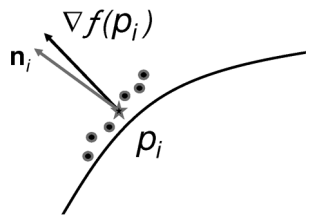
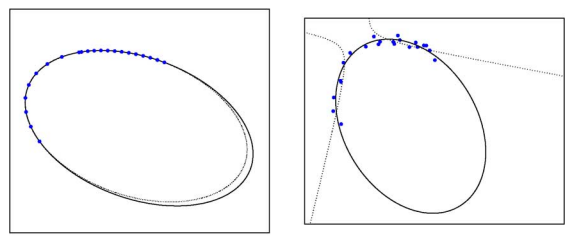


Fig. 3. Convergence criterion defined as the deviation between the IP normal and the local normal at each point.

TABLE I
PARAMETER SETUP

Initialization (the 3L alg.)		Optimization (LMA)		Stopping criterion	
δ	ϵ	λ	β	$\Delta\xi$	# iteration
0.1	1	0.01	0.5	<0.1	<25



(a) (b)

Fig. 4. Fitting a set of points from an ellipse. (a) Without noise: (dotted line) algebraic and (solid line) proposed methods. (b) Noisy data case: (dotted line) the algebraic approach misses the elliptic structure, whereas (solid line) the proposed approach reaches a good result.

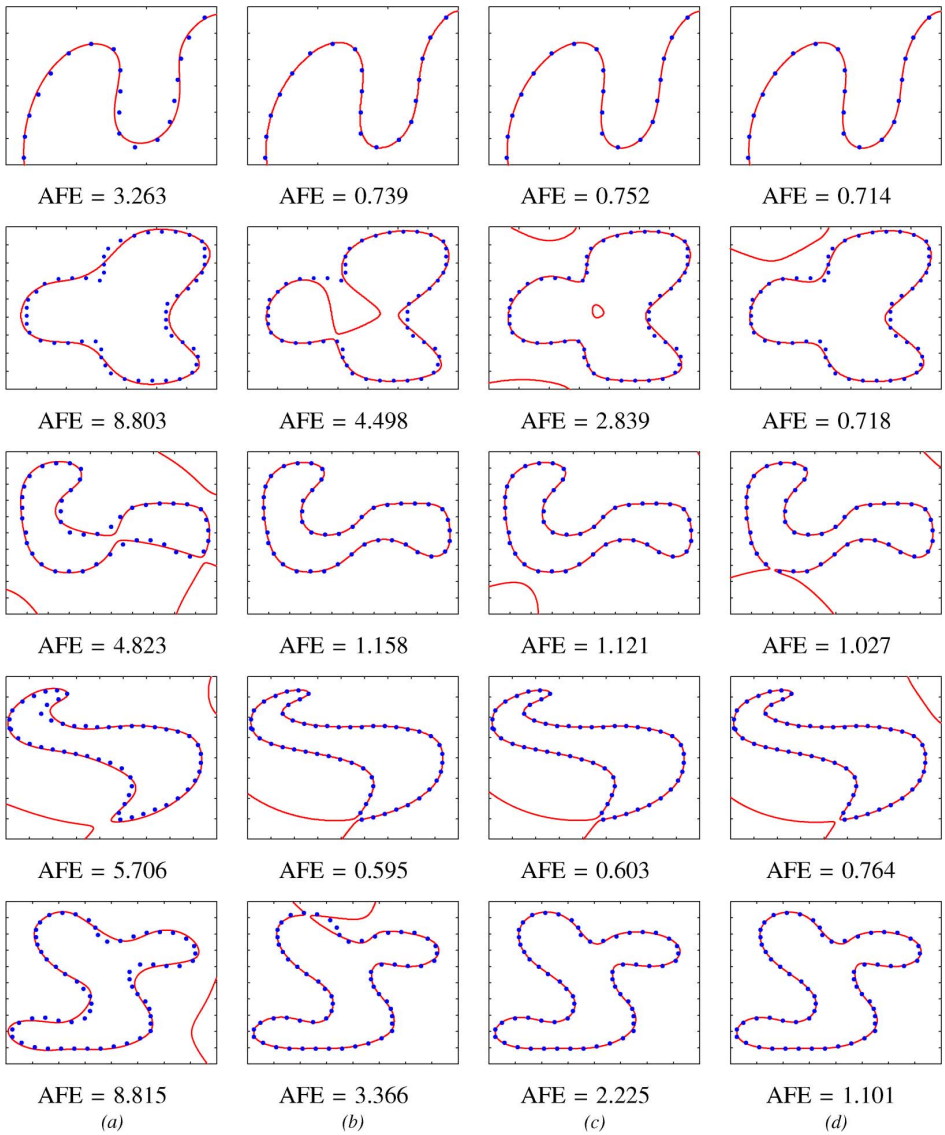


Fig. 5. Two-dimensional contours fitted by (first and second rows) fifth- and (third, fourth, and fifth rows) sixth-degree IPs' results from (a) 3L algorithm, (b) [10], (c) proposed approach, and (d) [16], which is used as a ground truth. AFE shows the accumulated fitting errors. The fourth row shows a case where [16] stops due to the maximum iteration criterion.

cosine inverse, is considered. Therefore, the criterion used for measuring the goodness of the current fitting result is

$$\xi(\mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N} 1 - \left|\frac{\boldsymbol{n}_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|}\right| \tag{24}$$

where N is the number of points in the original data set. LMA iterates while (24) decreases more than a user-defined threshold $\Delta\xi = \xi_t - \xi_{t-1}$ or a maximum number of iterations is reached.

IV. Experimental Results

The proposed method, which belongs to the geometric fitting category, is implemented and compared with the most important methods in the literature, both algebraic and geometric. The results presented here are evaluated using the *fitting error* (FE) computed for every single points with [16]. It is used to obtain a quantitative criterion for comparison, which is referred to as *accumulated FE* (AFE) where AFE $= \sum_{i=1}^{N} \text{FE}_i$. In all the cases, the given data points are centralized and scaled between $[-1, 1]$. The parameters of initialization (3L algorithm), optimization (LMA), and stopping criterion are empirically set up,

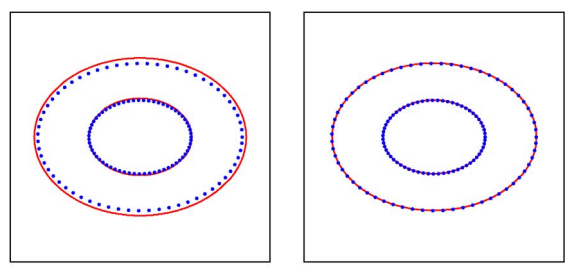


Fig. 6. Fitting two concentric ellipses. (a) Result from the 3L algorithm. (b) Result from the proposed approach.

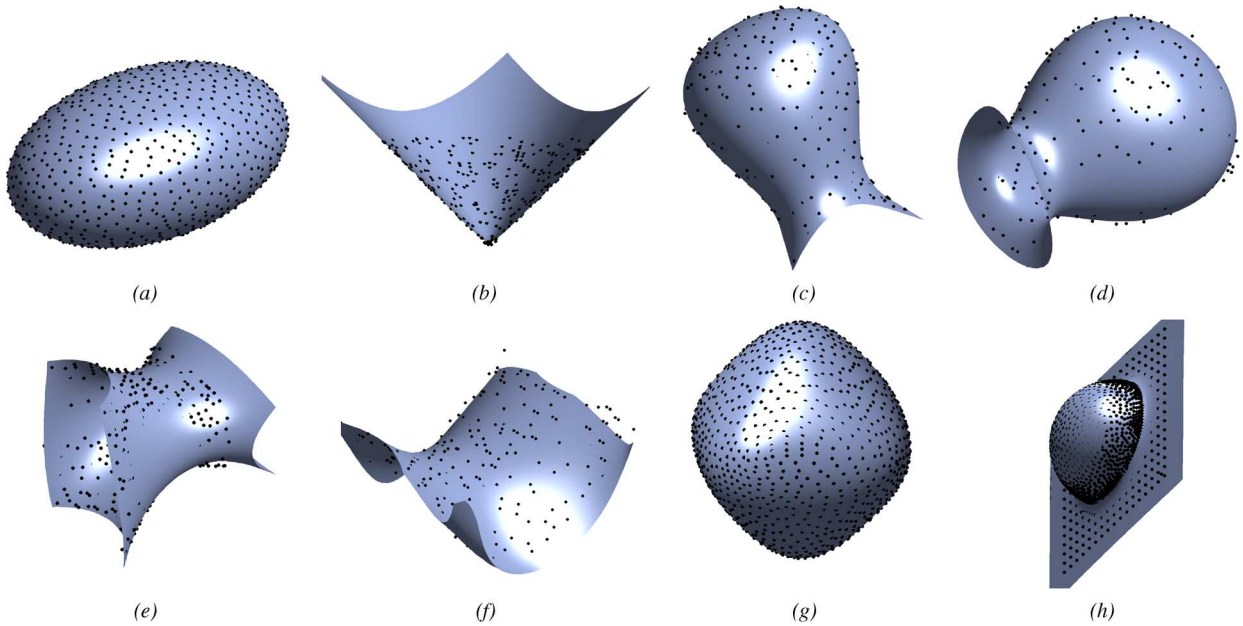


Fig. 7. Synthetic data sets fitted with the proposed approach.

as presented in Table I. The same initialization and stopping criterion have been used once the proposed approach is compared with other approaches.

In the 2-D case, different sets of points picked from quadric contours sampled with nonuniform distributions have been fitted with the proposed approach and compared with other approaches. Fig. 4(a) depicts the result of the proposed method when a nonuniformly distributed 2-D data set is fitted. Both the algebraic and proposed methods converge to a similar result, but problems arise when noise is added to the points. Fig. 4(b) highlights the robustness of the proposed method to noise, whereas the algebraic one misses the elliptic structure of the data and fits the patch as a split hyperbola. Fitzgibbon *et al.* [27] propose a fitting method just for 2-D elliptic cases based on algebraic approaches. From this simple example, one can understand the hardship for algebraic methods when the function space is bigger than the quadratic one.

The proposed approach is also implemented for fitting higher degree IPs. Fig. 5 shows 2-D contours fitted by fifth- and sixth-degree IPs (depending on the shape complexity) using the 3L algorithm [see Fig. 5(a)], the approach proposed in [10] [see Fig. 5(b)], the proposed approach [see Fig. 5(c)], and a nonlinear orthogonal-distance-based approach [16] used as a ground truth [see Fig. 5(d)]. The fitting error, computed over the whole set of points with [16], is used as a quantitative criterion for comparison. In all the cases, the accuracy obtained with the proposed approach considerably improves the one obtained with the 3L algorithm and, in most of the cases, gives better results than [10]; it is comparable (in one case better since the stopping criterion has been reached, see fourth row) to the results obtained when the nonlinear approach is used. Although out of the scope of this paper, it should be mentioned that in the 2-D case, the proposed approach is about 10 times faster than [16]. Finally, another challenging 2-D shape defined by two concentric ellipses has been fitted by a fifth-degree IP using the proposed approach; Fig. 6(a) shows the result from the 3L algorithm used as initialization of the proposed approach. The final result is depicted in Fig. 6(b).

TABLE II
SYNTHETIC DATA SET: AFE CORRESPONDING TO THE ILLUSTRATIONS PRESENTED IN Fig. 7

	IP degree	3L alg. [14]	Orthogonal fitting [16]	Prop. approach
Fig. 7(a)	second	9.58	5.56	**5.39**
Fig. 7(b)	second	2.42	1.32	**1.20**
Fig. 7(c)	fifth	1.89	0.69	**0.68**
Fig. 7(d)	third	1.93	1.73	**1.69**
Fig. 7(e)	fifth	2.67	1.28	**1.03**
Fig. 7(f)	third	3.80	**1.29**	1.31
Fig. 7(g)	fourth	2.20	**0.50**	0.51
Fig. 7(h)	third	1.17	0.42	**0.40**

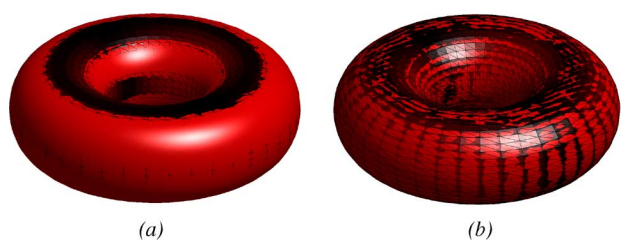


Fig. 8. Solid surface representing a fourth-degree IP; wire frame is used to visualize given data points. (a) IP obtained from the 3L algorithm. (b) Result from the proposed approach (note the similarity between the wire frame and the surface from the computed IP).

The proposed approach has been also evaluated with 3-D data sets, i.e., both synthetic and real data sets were fitted with low- and high-degree IPs. On average, in the 3-D case, the proposed approach is not as good as in the 2-D case, but it is about twice faster than [16]. Fig. 7 shows eight different results obtained with the proposed approach; in all the cases, the results are quite

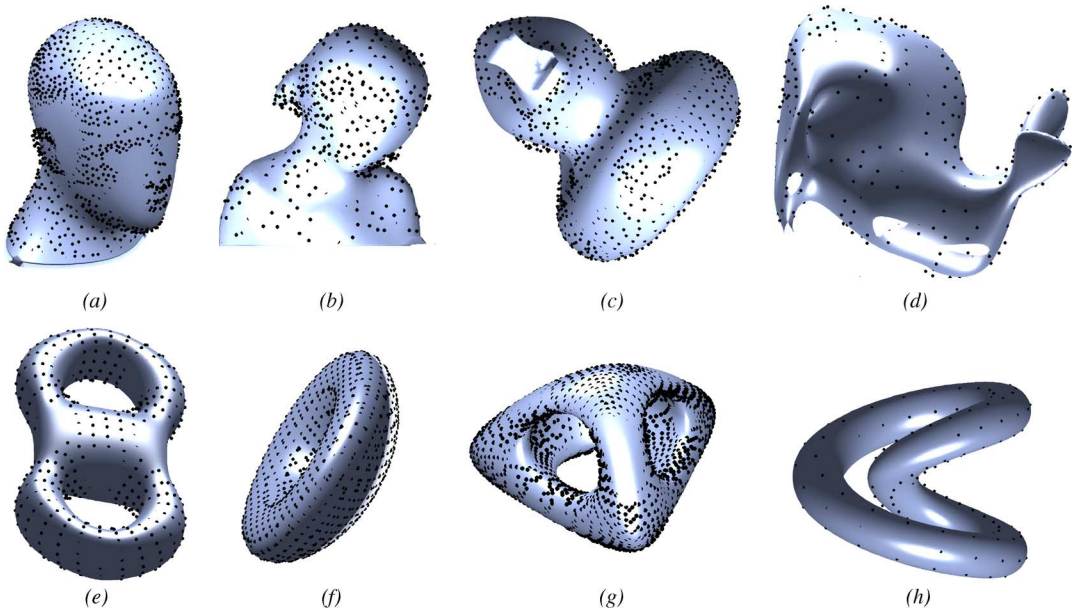


Fig. 9. Data set from AIM@SHAPE fitted with the proposed approach.

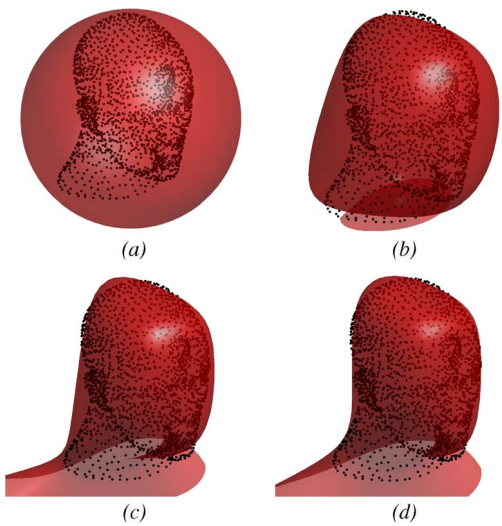


Fig. 10. (a) Fitting with a rough initialization. (b)–(d) First, second, and third iterations, respectively.

TABLE III
DATA SET FROM AIM@SHAPE: AFE CORRESPONDING TO THE ILLUSTRATIONS PRESENTED IN FIG. 9

	IP degree	3L alg. [14]	Orthogonal fitting [16]	Prop. approach
Fig. 9(a)	fourth	8.17	**5.31**	5.37
Fig. 9(b)	seventh	6.17	**5.76**	5.85
Fig. 9(c)	seventh	1.07	**0.56**	0.63
Fig. 9(d)	seventh	3.28	**1.51**	1.68
Fig. 9(e)	sixth	3.30	2.32	**2.27**
Fig. 9(f)	fourth	4.57	1.90	**1.88**
Fig. 9(g)	sixth	3.41	**2.92**	2.94
Fig. 9(h)	sixth	7.60	**6.61**	6.62

similar to the ones obtained with [16] and considerably better than those obtained with [14]. Table II presents the AFE obtained with the different approaches for a quantitative comparison. Note that these results were obtained once the stopping criterion has been reached; if a larger number of iterations are allowed, [16] achieves better results. The proposed algorithm has been tested with a more challenging 3-D data set with a different topology; Fig. 8 presents the results from both the 3L algorithm $(\text{AFE} = 0.06)$, which is used as an initialization of the proposed approach, and the final result obtained after ten iterations $(\text{AFE} = 1.00 \times 10^{-4})$. In this case, a fourth-degree IP has been used (solid surface), given that data points are represented by means of a wire frame just for a visual comparison.

In addition to the synthetic objects, a data set from AIM@SHAPE[2] has been used for evaluating the proposed approach. Fig. 9 shows eight illustrations of fourth-, sixth-, and seventh-degree IPs obtained with the proposed approach. Table III presents the AFE obtained with the different approaches for a quantitative comparison. Fig. 10 illustrates the independence to initial guess by using a sphere covering the given data set as an initialization [see Fig. 10(a)]. The first, second, and third iterations of the proposed approach are shown in Fig. 10(b)–(d), respectively; the result obtained after 25 iterations is already depicted in Fig. 9(a). Surface parameter refinements through these iterations are depicted in Fig. 11. Fig. 11(a) corresponds to the evolution of the 35 IP coefficients, whereas Fig. 11(b) shows how the AFE decreases with the iterations. Finally, Fig. 11(c) depicts the accumulated angle (23) used as a convergence criterion. It should be mentioned that this criterion has a similar behavior than Fig. 11(b), but its complexity is considerably lower.

V. CONCLUSION

This paper has presented a novel geometric approach for 2-D/3-D implicit polynomial fitting, which is based on a fast geometric distance estimation. Despite other geometric estimations, which are based on a single direction to find the foot

[2]http://shapes.aimatshape.net/

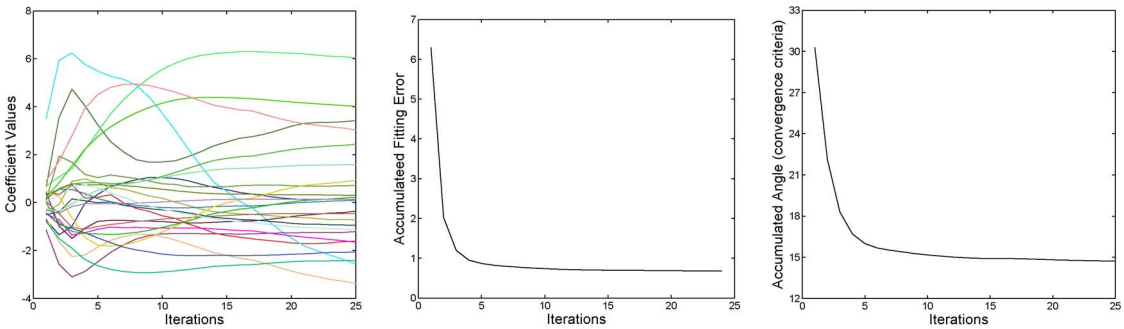


Fig. 11. Parameter evolution of Fig. 10 along 25 iterations: (a) IP coefficient values, (b) AFE, and (c) accumulated angle used as a convergence criterion.

point associated to each data point, the proposed one is based on two or three directions (depending on the data dimension). The smoothness and accuracy of the proposed distance have been shown. Additionally, the implicit connection between this distance and the IP coefficients has been presented and shown to be differentiable. This property allows the use of any gradient-based optimization techniques. In this paper, the LMA is applied to find the best set of surface parameters in an iterative way. Comparisons with state-of-the-art techniques are presented. Moreover, the proposed distance is proved to be a generalization of the distance presented in [10].

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous referees for their valuable and constructive comments.

REFERENCES

[1] M. Sarkis and K. Diepold, "Content adaptive mesh representation of images using binary space partitions," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1069–1079, May 2009.

[2] A. Jalba and J. Roerdink, "Efficient surface reconstruction from noisy data using regularized membrane potentials," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1119–1134, May 2009.

[3] C. Darolti, A. Mertins, C. Bodensteiner, and U. Hofmann, "Local region descriptors for active contours evolution," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2275–2288, Dec. 2008.

[4] V. Srikrishnan and S. Chaudhuri, "Stabilization of parametric active contours using a tangential redistribution term," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1859–1872, Aug. 2009.

[5] H. Ben-Yaacov, D. Malah, and M. Barzohar, "Recognition of 3-D objects based on implicit polynomials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 954–960, May 2010.

[6] B. Zheng, J. Takamatsu, and K. Ikeuchi, "An adaptive and stable method for fitting implicit polynomial curves and surface," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 561–568, Mar. 2010.

[7] R. Xu and M. Kemp, "Fitting multiple connected ellipses to an image silhouette hierarchically," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1673–1682, Jul. 2010.

[8] M. Rouhani and A. D. Sappa, "A novel approach to geometric fitting of implicit quadrics," in *Proc. 11th Int. Conf. Adv. Concepts Intell. Vis. Syst.*, Bordeaux, France, Sep. 28–Oct., 2 2009, pp. 121–132.

[9] A. Sappa and M. Rouhani, "Efficient distance estimation for fitting implicit quadric surfaces," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 3521–3524.

[10] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

[11] A. Helzer, M. Barzohar, and D. Malah, "Stable fitting of 2-D curves and 3-D surfaces by implicit polynomials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1283–1294, Oct. 2004.

[12] D. Keren and C. Gotsman, "Fitting curves and surfaces with constrained implicit polynomials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 1, pp. 476–480, Jan. 1999.

[13] T. Tasdizen, J. Tarel, and D. Cooper, "Improving the stability of algebraic curves for applications," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 405–416, Mar. 2000.

[14] M. Blane, Z. Lei, H. Civil, and D. Cooper, "The 3L algorithm for fitting implicit polynomials curves and surface to data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 3, pp. 298–313, Mar. 2000.

[15] M. Rouhani and A. D. Sappa, "Relaxing the 3L algorithm for an accurate implicit polynomial fitting," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, Jun. 2010, pp. 3066–3072.

[16] S. Ahn, W. Rauh, H. Cho, and H. Warnecke, "Orthogonal distance fitting of implicit curves and surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 620–638, May 2002.

[17] M. Aigner and B. Jütler, "Gauss–Newton-type technique for robustly fitting implicit defined curves and surfaces to unorganized data points," in *Proc. IEEE Int. Conf. Shape Model. Appl.*, New York, Jun. 2009, pp. 121–130.

[18] Y. Chen and C. Liu, "Quadric surface extraction using genetic algorithms," *Comput.-Aided Des.*, vol. 31, no. 2, pp. 101–110, Feb. 1999.

[19] P. Gotardo, O. Bellon, K. Boyer, and L. Silva, "Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 6, pp. 2303–2316, Dec. 2004.

[20] P. D. Sampson, "Fitting conic sections to very scattered data: An iterative refinement of the Bookstein algorithm," *Comput. Graph. Image Process.*, vol. 18, no. 1, pp. 97–108, Jan. 1982.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. 19th Annu. Conf. Comput. Graph. Interactive Techn., SIGGRAPH*, Chicago, IL, Jul. 1992, pp. 71–78.

[22] H. Helfrich and D. Zwick, "A trust region algorithm for parametric curve and surface fitting," *J. Comput. Appl. Math.*, vol. 73, no. 1/2, pp. 119–134, Oct. 1996.

[23] W. Wang, H. Pottmann, and Y. Liu, "Fitting B-spline curves to point clouds by curvature-based squared distance minimization," *ACM Trans. Graph.*, vol. 25, no. 2, pp. 214–238, Apr. 2006.

[24] D. Adi, S. Shamsuddin, and A. Ali, "Particle swarm optimization for nurbs curve fitting," in *Proc. IEEE Int. Conf. Comput. Graph., Imag. Vis.*, Tianjin, China, Aug. 2009, pp. 259–263.

[25] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1990.

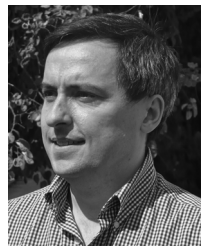
[26] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed. New York: Springer-Verlag, 2002.

[27] A. Fitzgibbon, M. Pilu, and R. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.

Mohammad Rouhani (S'09) received the Bachelor's degree in applied mathematics from the National University of Iran (currently Shahid Beheshti University), Tehran, Iran, in 2003 and the Master's degree from Sharif University of Technology, Tehran, in 2006. He is currently working toward the Ph.D. in computer vision at the Computer Vision Center, Barcelona, Spain.

Before studying for the Ph.D. degree, he spent two years in lecturing at Shomal University, Amol, Iran. His research is jointly on 3-D computer vision and computer graphics, including 2-D/3-D shape modeling, reconstruction, and registration.

Mr. Rouhani is a member of the Advanced Driver Assistance Systems Group, Computer Vision Center.



Angel Domingo Sappa (S'94–M'00) received the electromechanical engineering degree from the National University of La Pampa, General Pico, Argentina, in 1995 and the Ph.D. degree in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999.

In 2003, after holding research positions in France, U.K., and Greece, he joined the Computer Vision Center, Barcelona, where he is currently a Senior Researcher. His current research focuses on stereo image processing and analysis, 3-D modeling, and dense optical flow estimation. His research interests span a broad spectrum within the 2-D and 3-D image processing.

Dr. Sappa is a member of the Advanced Driver Assistance Systems Group, Computer Vision Center.