Implicit Polynomial Representation through a Fast Fitting Error Estimation

Mohammad Rouhani, *Student Member, IEEE* and Angel D. 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 surface (i.e., a triangle in 2D or a tetrahedron in 3D), 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 the current work, its use in a Levenberg-Marquardt framework is shown, which is specially devoted for non-linear 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 2D and 3D data sets, are provided. Comparisons with the state of the art are presented, showing the advantages of the proposed approach.**

***Index Terms*— Curve/Surface Fitting; Geometric Distance Estimation; Residual Error Minimization; Implicit Polynomial.**

I. Introduction

Implicit Polynomials (IPs) have been used in the computer vision field because they are advantageous compared with other representations. Firstly, they are a compact way to represent a given data set—in a 2D/3D space; and secondly, 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 in particular 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]). Implicit polynomial compactness has been also an attractive point to be exploited when a high level reasoning is needed (e.g., object recognition [5], object modelling [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) algebraic and ii) geometric; their difference depends on the criteria used to define "best" (i.e., accuracy vs speed). Next section briefly details these two approaches.

M. Rouhani and A. D. Sappa are with the Computer Vision Center, Campus UAB, 08193 Bellaterra, Barcelona Spain

The current 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 non-linear minimization framework: the Levenberg-Marquardt algorithm. The rest of the paper is organized as follows. Section 2 describes the problem and introduces related work. The proposed technique is presented in section 3. Section 4 gives experimental results and comparisons. Finally, conclusions are presented in section 5.

II. Problem Formulation and Related Work

The two major approaches in implicit polynomial fitting—*algebraic* and *geometric*—are briefly presented here to show the motivations of the proposed approach. Implicit polynomial 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:

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

where $\mathbf{c} = [c_{0,0,0}, c_{1,0,0}, ...c_{0,0,d}]^T$ is the column vector of polynomial coefficients having as many components as the combination of $(d+3)$ taken 3 at a time without repetitions: $C_3^{d+3} = \frac{(d+3)!}{d!3!}$; and $\mathbf{m}$ is the column vector of monomials: $\mathbf{m} = \mathbf{m}(\mathbf{x}) = [x^0y^0z^0, x^1y^0z^0, ...x^0y^0z^d]^T$; the fitting problem consists of first defining a criterion—or residual error—for measuring the closeness of the 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 2D or surfaces in 3D); then the fitting problem is defined as:

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

where $argmin_{\mathbf{c}}$ stands for the 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:

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

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: i) forcing the optimum vector to have a unit l_1 length (i.e., $\sum c_i = 1$), or ii) 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., $\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_{\mathbf{c}}^2(p_i)\|^2$=1). Due to the simplicity, the second normalization constraint is used in the current work, and the constant element in monomial, together with its corresponding coefficient, are removed in this case. This minimization problem is also equivalent to the overdetermined system of equations:

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

where M is the monomial matrix (every row contains the 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 computed algebraically through least squares solutions:

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

Two common problems, inherent to algebraic approaches, are (a) computational instability of the zero set; and (b) lack of geometric information of the data in this procedure. The non-iterative framework of algebraic approaches is an appealing feature for many applications. For instance, focussing on the instability problem, Helzer *et al.* [11] analyze the sensitivity of the zero set to small coefficient changes and minimize an upper bound of the error in order to have a more stable output. Keren *et al.* [12] try to constrain the surface parameter space in order to obtain a geometrically reasonable output. Tasdizen *et al.* [13] propose to add 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 [14] is a linear least squares polynomial fitting that consists of generating two additional *level sets*: $\Gamma_{-\delta}$ and $\Gamma_{+\delta}$ from the given data set Γ_0. These two additional data sets are generated so that one is internal and the other is external, and 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: $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$ the equation (5) could be represented by using a block matrix $\mathbf{M}_{3L}$ and a block column vector $\mathbf{b}$:

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

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

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

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

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

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 values ($\pm\epsilon$) are adjusted independently 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:

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

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

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

Point correspondence search: Regarding the first stage two different strategies have been proposed in the literature: (a) finding the shortest distance by solving a non-linear system (e.g., [16], [17]); and (b) computing an estimation of the shortest distance (e.g., [10], [18], [19]).

In [16] Ahn *et al.* propose 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 non-linear system of equations. Although this method

is precise enough, and even covers some well-known method in the literature like [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 l_1 and l_∞ norm 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 to approximate 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 principal components analysis (PCA) in a small $M \times M$ neighborhood centered at each point [21]. In other words, $\overrightarrow{n}_p = (n_1, n_2, n_3)$ is defined as the eigenvector of the local covariance matrix Cov associated with the smallest eigenvalue:

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

where $\tilde{\mathbf{p}} = \frac{1}{s}\sum_{i=1}^{s} p_i$ is the vector showing the mean position of the neighboring points in the $M \times M$ 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 $\overrightarrow{n}_p$:

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

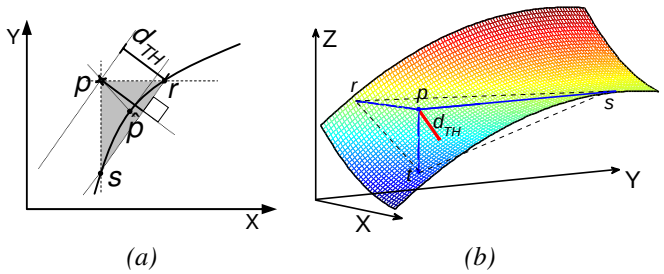
The intersection is used as an approximation for the foot-point $\hat{p}$ in (10).

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:

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

This approximated distance is used in an iterative weighted least squares method as well as in a nonlinear optimization framework. In addition, a new constraint is imposed on the coefficient vector, which is based on the data points as well as 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, and the best set of parameters will be found.

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], Particle Swarm [24]) in the current work the Levenberg-Marquardt algorithm (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 LMA are given in Section III-B).



(a) (b)

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

III. Proposed Approach

The current work proposes a geometric approach to tackle IPs fitting through an estimation of the orthogonal distance. In spite of being focussed on the geometric framework, the polynomial coefficients are firstly initialized by using an algebraic based algorithm—the 3L algorithm [14] has been used. 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. 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 till convergence is reached; they are detailed next.

A. Approximated Residual Error

The first contribution of the current work lies in a direct approach to estimate the orthogonal distance. It works as follows. First a *simplex* is constructed through each point and its intersections along the coordinate axis. A simplex is a triangle in 2D and a tetrahedron in 3D, as depicted in Fig. $1(a)$ and Fig. $1(b)$ respectively. Without loss of generality, the 3D 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 work, since the fitted curve/surface is defined by the implicit polynomial (1), the intersecting points are found by computing the closest root of a one dimensional function 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 be the three intersections with the current surface,

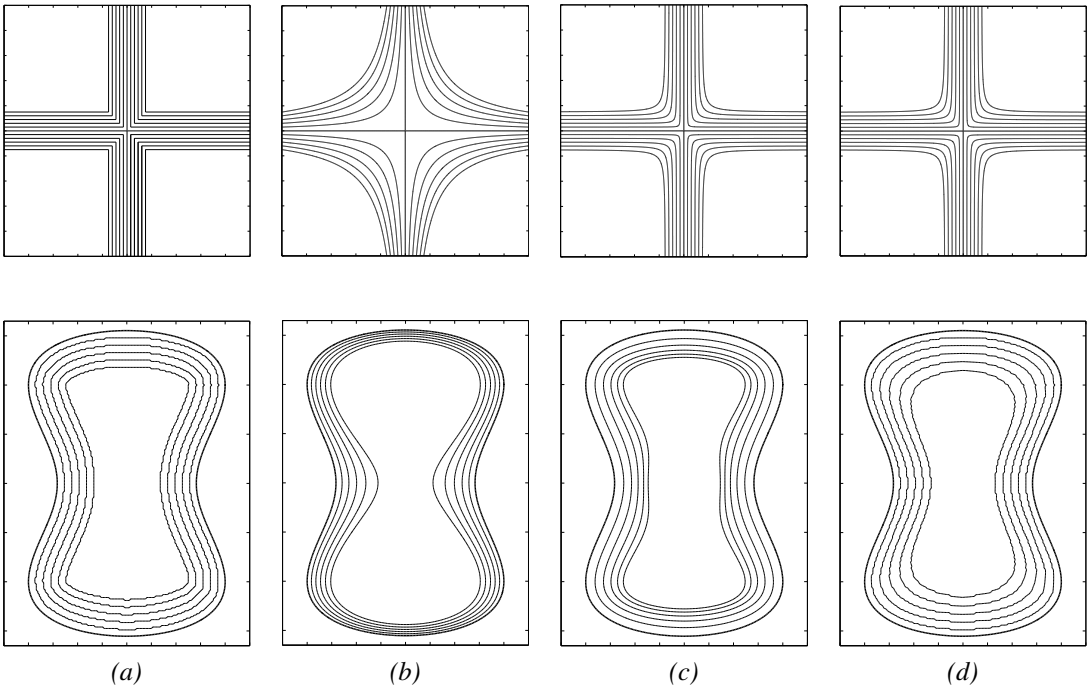


Fig. 2. Contour of constant distance for: (a) Orthogonal distance; (b) algebraic distance; (c) [10]; (d) proposed distance estimation.

which 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 easily be computed from the following relationship:

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

where $\times$ refers to the cross product operator between two vectors. Similar relationship can be obtained in the 2D 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 at 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 case 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, the 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 the point p_i could be approximated as follows:

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

thus, the proposed distance is a generalization of the 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 the orthogonal one. The first row of the figure shows the iso-contours [1] of the set $\{(x, y) : xy = 0\}$, which consists of two intersecting lines, and the second row shows iso-contours of a regular curve $\{(x, y) : 8x^2 + (y^2 - 4)^2 - 32 = 0\}$. As illustrated in last two columns, our method and Taubin's behave similarly 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 have a quite similar result to the iso-contours 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. The accumulation of all these distances provides a good criteria for curve surface fitting:

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

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

This distance is in the least squares form where each term is non-linear with respect to the 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 RANSAC based framework to find the quadratic surface parameters [9]. Other optimization techniques, like Genetic Algorithm (GA) [19] or Quasi-Newton method [23], have been already used in surface fitting.

The current work not only propose a simple and fast distance estimation approach but also, as a second contribution, it shows how this estimation can be used in a non-linear framework. In the current work, the Levenberg-Marquardt algorithm (LMA) has been used [25] to optimize the distance (19) with respect to the curve/surface parameters. LMA is specifically designed for non-linear 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 mentioned above the result from the 3L algorithm has been used as initialization). LMA updates these parameters iteratively as follows:

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

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

Each iteration of LMA contains two stages: 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 one dimensional function with respect to t. The direction vector $\mathbf{d}$ is set to $e_1 = (1,0,0)^T$, $e_2 = (0,1,0)^T$ and $e_3 = (0,0,1)^T$ 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 mentioned above, it should be noticed here that if we stop the Newton's method after one iteration, the proposed distance will be computed easily 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 the parameter vector $\mathbf{c}$. The Jacobian matrix could be directly derived through the differentiation rules as follow:

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

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, the term $|pr|$ is computed just by considering its x-component: $|pr| = (r_i^x - p_i^x)$ 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 *j*-th monomial component calculated in the intersection. The 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 algorithm iterates equations (20) till convergence is reached. In the current work convergence criteria has been defined using the deviation between the IP normal and the local normal at each point (see illustration in Fig. 3). This criteria, 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). So the only required computation is regarding the angle estimation:

$$\theta_i = cos^{-1}\left(\ \frac{n_i.\nabla f(p_i)}{\|\nabla f(p_i)\|}\ \right), \tag{23}$$

additionally, since $cos^{-1}|_{[0,1]\rightarrow[0,pi/2]}$ is monotonic, just the absolute value of the inner expression, without calculating the cosine inverse, is considered. Therefore the criterion used for measuring the goodness of the current fitting result is:

$$\xi(\mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N} 1 - \left|\ \frac{n_i.\nabla f(p_i)}{\|\nabla f(p_i)\|}\ \right|, \tag{24}$$

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.

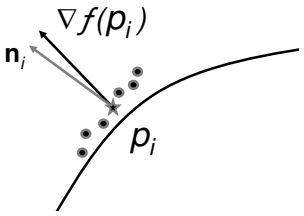


Fig. 3. Convergence criteria defined as the deviation between the IP normal and the local normal is each point.

IV. Experimental Results

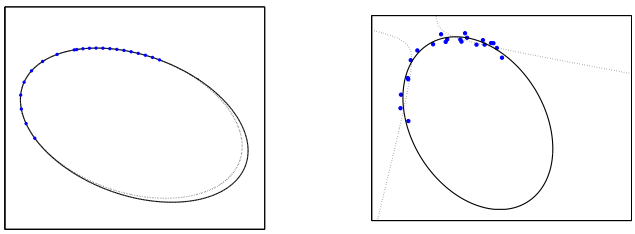
The proposed method, which belongs to the geometric fitting category, is implemented and compared with the most important methods in the literature, both algebraic and geometric. The results presented in this section are evaluated using the *fitting error* (FE) computed for every single points with [16]. It is used to obtain a quantitative criterion for comparison, which is referred to as *Accumulated Fitting Error* (AFE): AFE= $\sum_{i=1}^{N} FE_i$. In all the cases the given data points are centralized and scaled between [-1,1]. The parameters of

initialization (3L algorithm), optimization (LMA) and stoping criterion are empirically set up as presented in Table I. The same initialization and stopping criterion have been used once the proposed approach is compared with other approaches.

TABLE I
PARAMETERS SET UP.

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

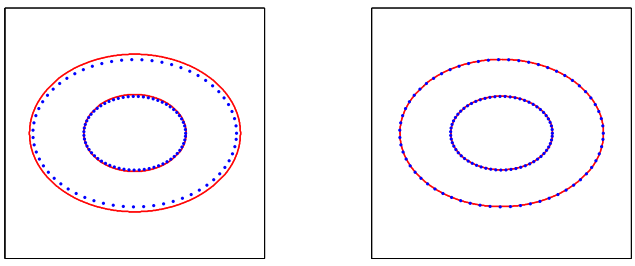
In the two dimensional case, different sets of points picked from quadric contours sampled with non-uniform distributions have been fitted with the proposed approach and compared with other approaches. Fig. 4(a) depicts the result of the proposed method when a non-uniformly distributed 2D data set is fitted. Both algebraic and proposed method 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 2D 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.



(a) *(b)*

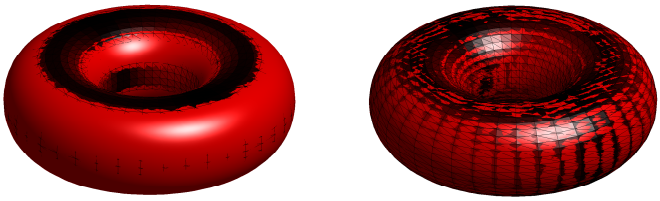
Fig. 4. Fitting a set of points from an ellipse. (a) Without noise: Algebraic (dotted line) and proposed method (solid line). (b) Noisy data case: Algebraic (dotted line) misses the elliptic structure, while the proposed approach (solid line) reaches a good result.

The proposed approach is also implemented for fitting higher degrees IPs. Figure 5 shows 2D contours fitted by fifth and sixth degree IPs (depending on the shape complexity)



(a) *(b)*

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



(a) *(b)*

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 wire frame and the surface from the computed IP).

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**

using: the 3L algorithm (Fig. 5(a)); the approach proposed in [10] (Fig. 5(b)); the proposed approach (Fig. 5(c)); and a non-linear orthogonal distance based approach [16] used as a ground truth (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]; actually, it is comparable (in one case better since the stopping criteria has been reached, see fourth row) to the results obtained when the non-linear approach is used. Although out of the scope of the current work, it should be mentioned that in the 2D case the proposed approach is about ten times faster than [16]. Finally, another challenging 2D shape defined by two concentric ellipses has been fitted by a fifth degree IP using the proposed approach; Figure 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).

The proposed approach has been also evaluated with 3D data sets, both synthetic and real data sets were fitted with low and high degree IPs. On average, in the 3D case, the proposed approach is not as good as in the 2D case, but it is about twice faster than [16]. Figure 7 shows eight different results obtained with the proposed approach; in all the cases the results are quite similar to the ones obtained with [16], and considerably better than those obtained with [14]. Table II presents the Accumulated Fitting Error obtained with the different approaches for a quantitative comparison. Note that

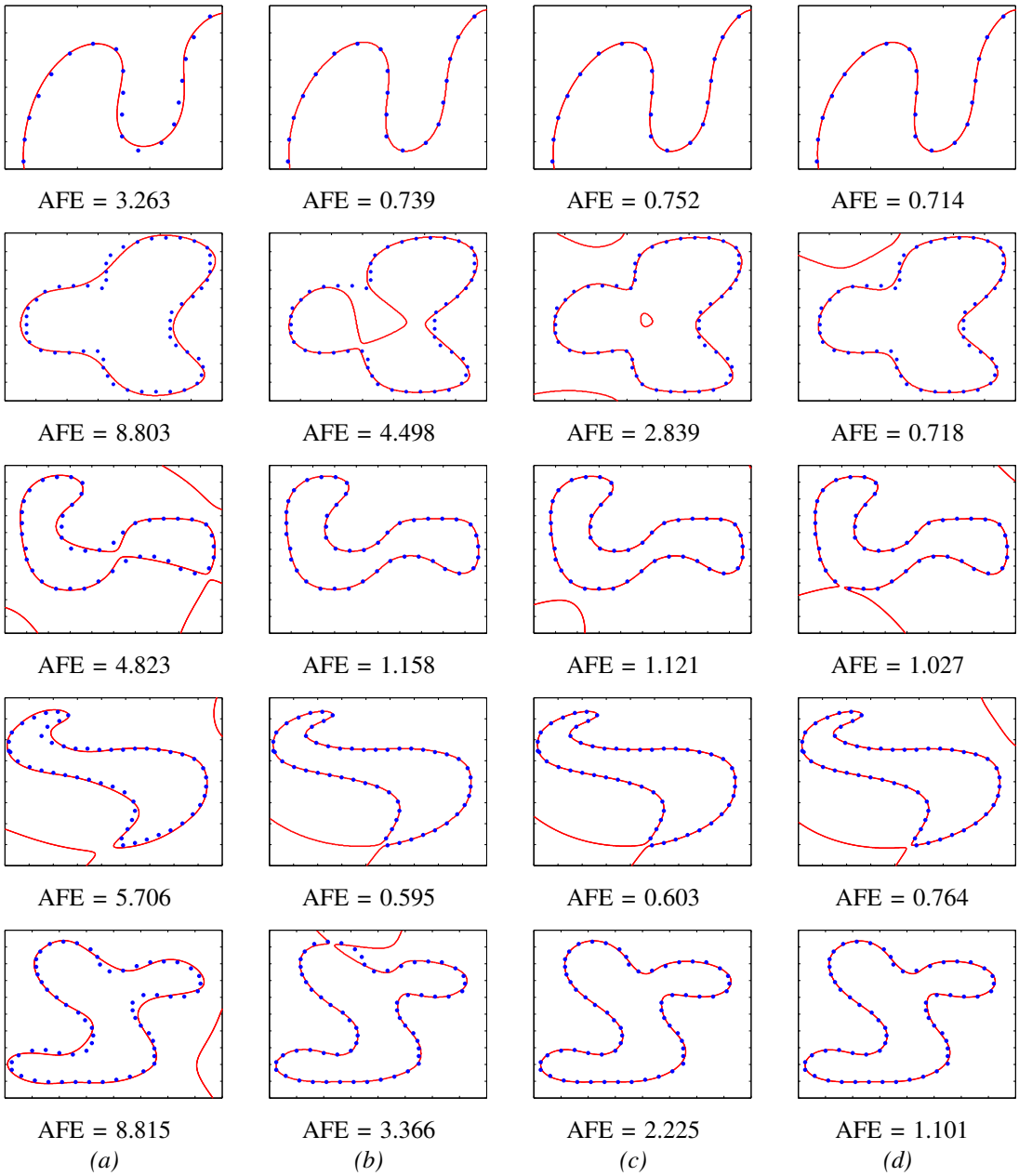


Fig. 5. 2D contours fitted by fifth (1st and 2nd rows) and sixth (3rd, 4th and 5th rows) degree IPs, results from: (a) the 3L algorithm; (b) [10]; (c) proposed approach; (d) [16], which is used as ground truths. AFE shows the accumulated fitting error respectively. The fourth row shows a case where [16] stops due to the maximum iteration criterion.

these results were obtained once the stopping criteria has been reached; if larger number of iterations are allowed, [16] achieves better results. The proposed algorithm has been tested with a more challenging 3D data set with a different topology; Figure 8 presents results from both the 3L algorithm (AFE=0.06), which is used as an initialization of proposed approach, and the final result obtained after 10 iterations (AFE=1.00×10^{-4}). In this case a fourth degree IP has been used (solid surface); given 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. Figure 9 shows eight illustrations of fourth, sixth and seventh degree IPs obtained with the proposed approach. Table III presents the Accumulated Fitting Error obtained with the different approaches for a quantitative comparison. Figure 10 illustrates the independence to initial guess by using a sphere covering the given data set as an initialization (see Fig. $10(a)$). First, second and third iterations of the proposed approach are shown in Fig. $10(b)$, (c) and (d) respectively; result obtained after 25 iterations is already depicted in Fig. $9(a)$. Surface parameter refinements through these iterations are depicted in Fig. 11. Figure $11(a)$ corresponds to the evolution of the 35 IP coefficients, while Fig. $11(b)$ shows how the AFE decreases with the iterations. Finally, Fig. $11(c)$ depicts the accumulated angle (23) used as a convergence

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

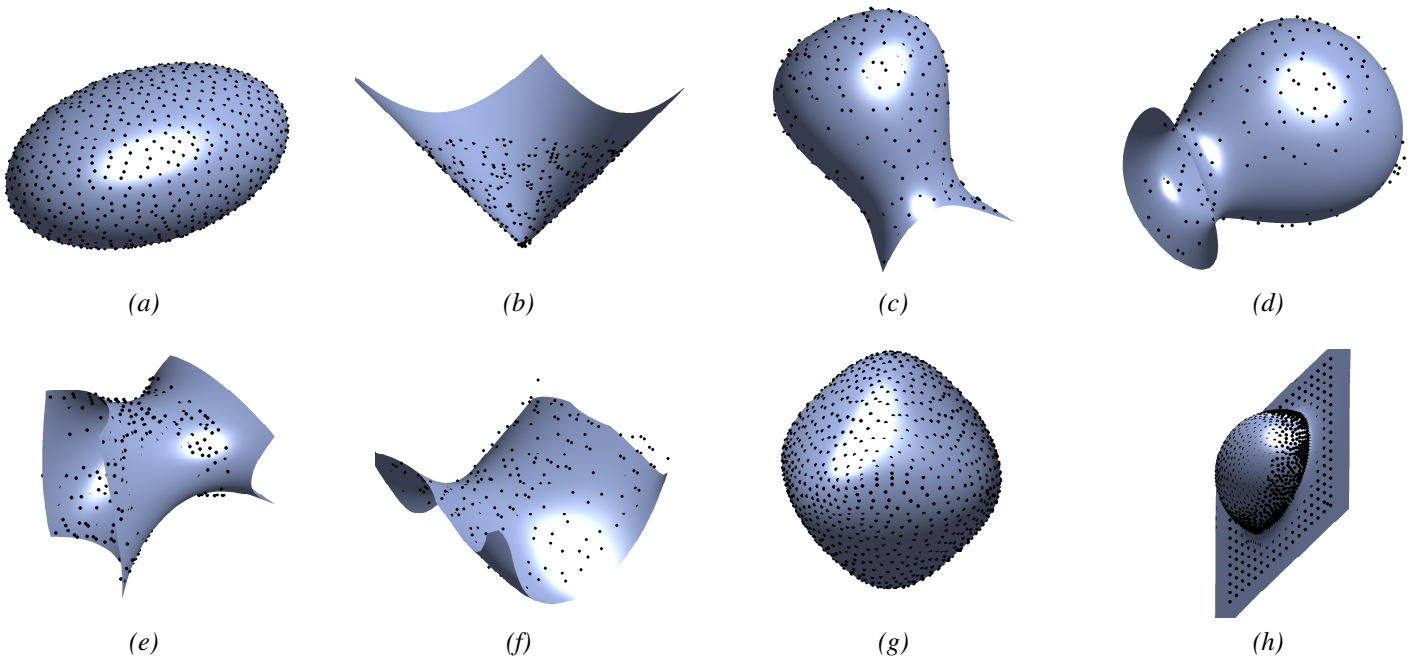


Fig. 7. Synthetic data sets fitted with the proposed approach.

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

criteria. It should be mentioned that this criteria has a similar behavior than Fig. 11(b) but its complexity is considerably lower.

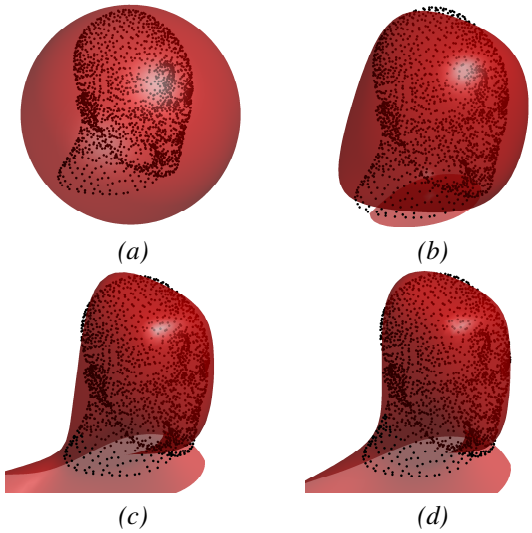


Fig. 10. (a) Fitting with a rough initialization. (b), (c) and (d) First, second and third iterations respectively.

V. CONCLUSIONS

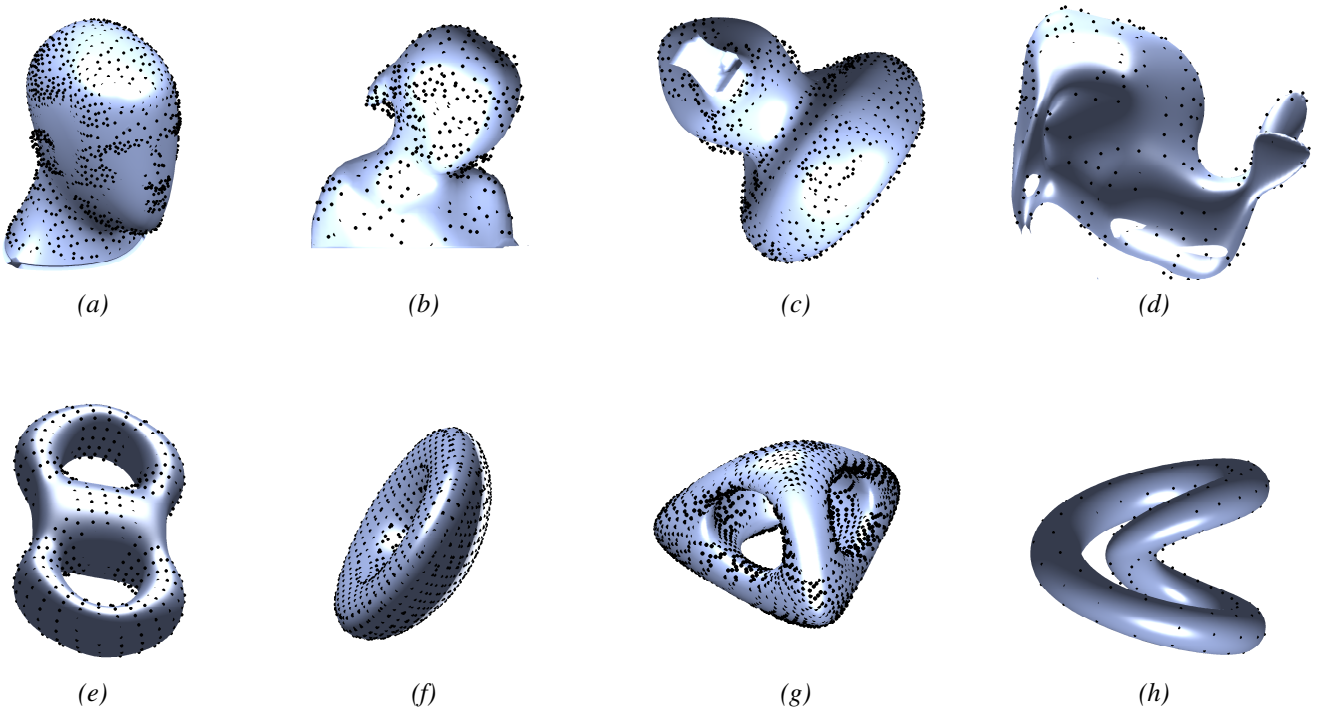
This paper presents a novel geometric approach for 2D/3D implicit polynomial fitting, which is based on a fast geometric distance estimation. Despite other geometric estimations, which are based on a single direction to find the foot-point associated to each data point, the proposed one is based on two 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 the current work the Levenberg-Marquardt algorithm 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. This work was partially supported by the Government of Spain under research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and MICINN projects TIN2011-25606 and TIN2011-29494-C03-02.

REFERENCES

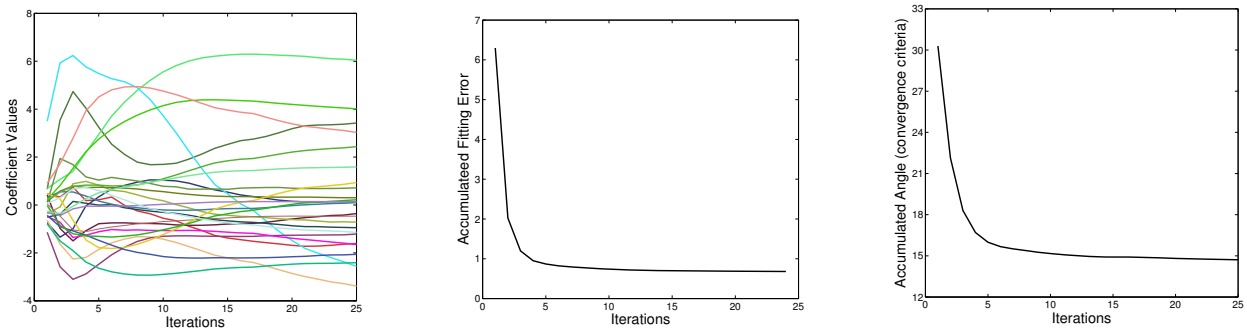
[1] M. Sarkis and K. Diepold, "Content adaptive mesh representation of images using binary space partitions,"



(a) *(b)* *(c)* *(d)*

(e) *(f)* *(g)* *(h)*

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



(a) *(b)* *(c)*

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

IEEE Trans. on Image Processing, 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. on Image Processing*, 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. on Image Processing*, vol. 17, no. 12, pp. 2275–2288, December 2008.

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

[5] H. Ben-Yaacov, D. Malah, and M. Barzohar, "Recognition of 3d objects based on implicit polynomials," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 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. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 561–568, March 2010.

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

[8] M. Rouhani and A. D. Sappa, "A novel approach to geometric fitting of implicit quadrics," in *Proc. of the 11th Int. Conf. on Advanced Concepts for Intelligent Vision Systems*, Bordeaux, France, Sept. 28–Oct. 2 2009, pp. 121–132.

[9] A. Sappa and M. Rouhani, "Efficient distance estimation for fitting implicit quadric surfaces," in *Proc. of the IEEE Int. Conf. on Image Processing*, Cairo, Egypt, November 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. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, November 1991.

[11] A. Helzer, M. Barzohar, and D. Malah, "Stable fitting of 2d curves and 3d surfaces by implicit polynomials," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1283–1294, October 2004.

[12] D. Keren and C. Gotsman, "Fitting curves and surfaces with constrained implicit polynomials," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 1, pp. 476–480, January 1999.

[13] T. Tasdizen, J. Tarel, and D. Cooper, "Improving the stability of algebraic curves for applications," *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 405–416, March 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. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 298–313, March 2000.

[15] M. Rouhani and A. D. Sappa, "Relaxing the 3L algorithm for an accurate implicit polynomial fitting," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, San Francisco, USA, June 2010, pp. 3066–3072.

[16] S. Ahn, W. Rauh, H. Cho, and H. Warnecke, "Orthogonal distance fitting of implicit curves and surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 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. of the IEEE Int. Conf. on Shape Modelling and Application*, New York, USA, June 2009, pp. 121–130.

[18] Y. Chen and C. Liu, "Quadric surface extraction using genetic algorithms," *Computer-Aided Design*, vol. 31, no. 2, pp. 101–110, February 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. on Systems, Man, and CyberneticsPart B: Cybernetics*, vol. 34, no. 6, pp. 2303–2316, December 2004.

[20] P. D. Sampson, "Fitting conic sections to very scattereda data: An iterative refinement of the bookstein algorithm," *Computer Graphics and Image Processing*, vol. 18, no. 1, pp. 97–108, January 1982.

[21] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. of the 19th Annual Conf. on Computer Graphics and Interactive Techniques, SIGGRAPH 1992*, Chicago, IL, USA, July 1992, pp. 71–78.

[22] H. Helfrich and D. Zwick, "A trust region algorithm for parametric curve and surface fitting," *Journal of Computational and Applied Mathematics*, vol. 73, no. 1-2, pp. 119–134, October 1996.

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

[24] D. Adi, S. Shamsuddin, and A. Ali, "Particle swarm optimization for nurbs curve fitting," in *Proc. of the IEEE Int. Conf. on Computer Graphics, Imaging and Visualization*, Tianjin, China, August 2009, pp. 259–263.

[25] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, second edition, 1990.

[26] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. Springer, third edition, 2002.

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

Mohammad Rouhani (S'09) received his bachelor degree in applied mathematics from the National University of Iran (Shahid Beheshti), in 2003, and his master degree from Sharif University of Technology in Tehran, Iran, in 2006. After two years of lecturing at Shomal University, Amol, Iran, he joined the Computer Vision Center in Barcelona, Spain, where he is currently doing his Ph.D on Computer Vision. He is a member of the Advanced Driver Assistance Systems Group. His research is jointly on 3D Computer Vision and Computer Graphics, including 2D/3D shape modelling, reconstruction and registration. He is a student member of the IEEE.

Angel Domingo Sappa (S'94-M'00) received the electromechanical engineering degree from 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, the UK, and Greece, he joined the Computer Vision Center, where he is currently a senior researcher. He is a member of the Advanced Driver Assistance Systems Group. His research interests span a broad spectrum within the 2D and 3D image processing. His current research focuses on stereoimage processing and analysis, 3D modeling, and dense optical flow estimation. He is a member of the IEEE.