A Novel Approach to Geometric Fitting of Implicit Quadrics*

Mohammad Rouhani and Angel D. Sappa

Computer Vision Center, Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{rouhani,asappa}@cvc.uab.es

Abstract. This paper presents a novel approach for estimating the geometric distance from a given point to the corresponding implicit quadric curve/surface. The proposed estimation is based on the height of a tetrahedron, which is used as a coarse but reliable estimation of the real distance. The estimated distance is then used for finding the best set of quadric parameters, by means of the Levenberg-Marquardt algorithm, which is a common framework in other geometric fitting approaches. Comparisons of the proposed approach with previous ones are provided to show both improvements in CPU time as well as in the accuracy of the obtained results.

1 Introduction

Fitting a curve or surface to a given cloud of points is a fundamental problem in computer vision and geometric modelling. It has been an active topic during the last two decades [1,2]. The appearance of new sensors, which allow to obtain a large amount of 3D data in a reduced time, and the need to process all this information efficiently have opened new challenges looking for efficient fitting approaches.

Although there are many tools in Computer Aided Design to represent curves and surfaces, the implicit representation is more efficient since it avoids the parametrization problem (i.e., difficulties arise especially when one should face up to unorganized cloud of points). The surface is described as the set of points $\mathbf{X}$ satisfying the equation $f(\mathbf{c}, \mathbf{X}) = 0$; where $\mathbf{c}$ is the set of parameters. This set of points is also referred as *zero set* of f. Although different function spaces could be used for the implicit representation (e.g., B-Spline functions [2,3], and radial basis functions [4]) in the current work the implicit polynomial case is considered.

Having represented the surface as an implicit polynomial, the fitting problem can be modelled as an optimization problem: *finding the set of parameters that minimize some distance measures between the given set of points and the fitted*

* This work has been partially supported by the Spanish Government under project TRA2007-62526/AUT; research programme Consolider-Ingenio 2010: MIPRCV (CSD2007-00018); and Catalan Government under project CTP 2008ITT 00001.

J. Blanc-Talon et al. (Eds.): ACIVS 2009, LNCS 5807, pp. 121–132, 2009.

curve or surface. The most natural way to define the distance is to measure the deviation of the function values from the expected value (i.e., zero) at each given point. This measure criterion is referred in the literature as *algebraic distance* [5,6].

Another distance measure, referred as *orthogonal* or *geometric distance*, is defined as the shortest distance from the given point to the fitting surface. On the contrary to the algebraic distance, this distance has a direct link to the geometry of the data, and its final result makes sense as a consequence. Although this definition of the distance is complete and leads us to the best fitting result, it has a nonlinear nature with the model parameter, that discourages its use. On the other hand, since there is no closed formula to compute the shortest distance, two strategies have been proposed in the literature: (a) compute the orthogonal distance by means of an iterative approach (e.g., [2,7]); and (b) compute an approximation to this distance and use it as the residual value of the point (e.g., [8,9]).

Algebraic and geometric distances are two different viewpoints for the fitting problem. Although both of them could be exploited for some optimization models leading to the optimal parameters in their own sense, the frameworks they use are different. Algebraic fitting methods are based on quadratic optimization model (least square) giving a *non-iterative* unique solution, while the geometric ones are based on some non-linear models giving the solution through *iterative* algorithms.

The current work proposes a novel technique for an efficient estimation of the geometric distance, from a given point p to the corresponding implicit quadric fitting surface. This distance is computationally efficient as well as a reliable approximation. This geometric criterion is later on used in an optimization framework. 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, conclusion and future work are presented in section 5.

2 Related Work

In this section, the two major approaches in surface fitting: *algebraic* and *geometric*, are presented in more details to show the motivations of the proposed approach. Furthermore, a brief introduction to the optimization method used in the current work is also given.

Fitting problems aim at fining a curve or surface *close* to a given cloud of points $\mathbf{X} = \{p_i\}_{i=1}^{n}$. Before explaining the meaning of *close* we should define the implicit surface we want to find. Without loss of generality let us consider the quadratic implicit surfaces:

$$\begin{aligned} f_{\mathbf{c}}(x, y, z) = c_1x^2 + c_2y^2 + c_3z^2 + c_4xy + c_5xz \\ +c_6yz + c_7x + c_8y + c_9z + 1 = 0, \end{aligned} \tag{1}$$

where $\mathbf{c} = (c_1, ..., c_9)^T$ is the vector we are searching for. This implicit representation provides us many facilities; e.g., for closed surfaces we can easily find out whether a point is inside or outside of the surface just through checking the sign of f at the given point.

2.1 Algebraic Approaches

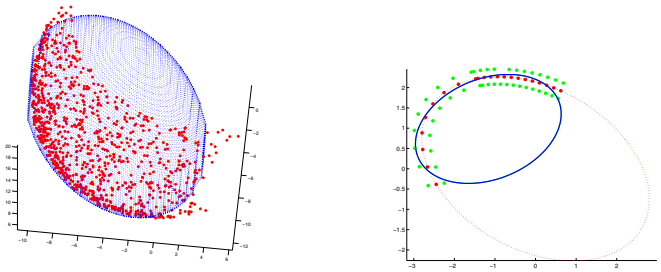
Since the implicit representation is used, a point is on the surface if and only if the output of $f_{\mathbf{c}}$ in (1) is zero at the given point. It leads us to define the following optimization criterion, which is known as algebraic approach:

$$\sum_{X} f_{\mathbf{c}}^2(x, y, z). \tag{2}$$

This minimization problem is also equivalent to the overdetermined system $M\mathbf{c} = \mathbf{b}$, where M is the monomial matrix computed at given points, and $\mathbf{b}$ is a column vector containing -1. Regardless to these different viewpoints, the optimal solution could be computed through least square solutions:

$$\mathbf{c} = (M^T M)^{-1} M^T b. \tag{3}$$

Algebraic distance has a simple formulation and a straightforward solution that is not iterative. Unfortunately, this method could fail for real world data set, where there is no information about the distribution of noise. As an illustration Fig. 1(a) shows how this method, despite its simplicity, fails to fit a cloud of points[1] picked from a spherical patch. Here the algebraic method tries to put the value of implicit function close to zero, and because of this *algebraic criterion* the curvature of the data, which is an important *geometric* property, is missed.



(a) *(b)*

Fig. 1. (*a*) A cloud of 3D real data fitted through the simple algebraic approach (3). Some clusters of points are inside the surface and some outside. (*b*) Result of the 3L algorithm [6] (solid line). Since the data points are not uniformly distributed it fails to fit them; while, in this case, the simple algebraic (3) is able to give the exact solution (dotted line).

Two common problems, inherent to algebraic approaches, are (*a*) computational instability of the zero set; and (*b*) lack of geometric sense in this procedure. However, the non-iterative framework of algebraic approaches has attracted the attention of many researchers. For instance, focussing on the instability problem, Hezer et al. [5] analyze the sensitivity of the zero set to small coefficient

[1] K2T structured light camera system, University of South Florida.

changes and minimize an upper bound of the error in order to have a more stable output. In addition to the instability problem, algebraic method does not have any geometric meaning. Indeed what is happening inside the procedure is just to minimize a quadratic optimization function—which is equivalent to an overdetermined system of equations. Keren et al. [10] try to constrain the surface parameter space in order to obtain a geometrically reasonable output. Blane et al. [6] and Tasdizen et al. [11] in parallel researches propose to add some geometric concept inside the optimization problem.

In [11] they try to maintain the estimated gradient value at each data points while they fit the data. In [6] Blane et al. add two complementary sets to the given data set, which are a shrunken and an expanded version of the original data set (this method is referred as 3L algorithm, which stands for 3 level set). After finding these complementary sets through local regression, they try to find an implicit function not only gaining zero in the original data set, but achieving +1 and -1 respectively in the shrunken and expanded sets. Although it is a robust and widely used approach in the literature, it could fail for some simple cases even in 2D. Figure 1(b) shows a set of non-uniformly distributed points on an ellipse. Because of two other supplementary sets, the 3L algorithm tried to find a compromise between all three sets, so fails to fit the right ellipse. In the figure the dotted line ellipse corresponds to the output of 3L, while the solid line one is the output of simple algebraic (3), which manages to fit it.

2.2 Geometric Approaches

In addition to algebraic methods, there is another category based on geometric distance—or an approximation of it—usually referred as *geometric approach*. 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. Thus, in general case of geometric methods we have the following optimization problem:

$$min_{\mathbf{c}}(\sum_{i=1}^{n} min_{\hat{p}_i} d(p_i, \hat{p}_i)), \tag{4}$$

where each $\hat{p}_i$ is the correspondence of p_i on the surface.

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 refine it till convergence is reached. So, the fitting problem is split up into two stages: 1) point correspondence search; and 2) surface parameter refinement. The first stage deals with the inner part of (4), while the second one concerns about the outer one.

Point correspondence search: Regarding to the first stage, we need to find the correspondence for each data point. For this purpose, two different strategies have been proposed in the literature: (a) finding the shortest distance by solving a non-linear system (e.g., [2,7]); and (b) computing an estimation of the shortest distance (e.g., [8,9]).

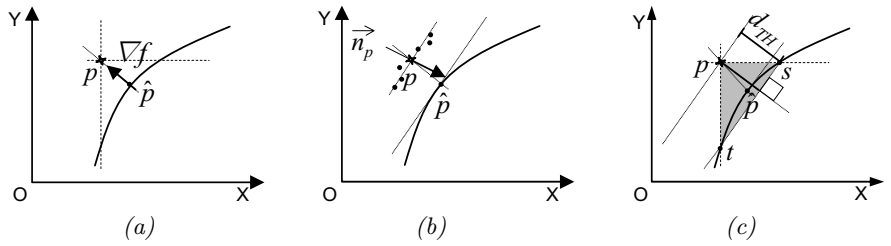


Fig. 2. Orthogonal distance estimation between p and a quadric curve. (a) The shortest distance is found using the iterative method proposed by [7]. (b) An estimation of the curve orientation, through applying PCA in a local neighborhood, used by [9]. (c) Distance estimation based on the proposed approach (i.e., by finding the Triangle Height, d_{TH}).

In [7] Ahn et al. propose the *direct 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 $f(\hat{p}) = 0$. Furthermore, the line connecting the data point with the foot-point must be parallel to the ∇f at the foot-point, where ∇ is the gradient operator (see Fig. 2(a)). In other words, the equation $\nabla f \times (\hat{p} - p) = 0$ must be satisfied. Merging these two conditions, the following system of equations must be solved:

$$\begin{pmatrix} f_{\mathbf{c}} \\ \nabla f_{\mathbf{c}} \times (\hat{\boldsymbol{p}} - \boldsymbol{p}) \end{pmatrix} = \mathbf{0}. \tag{5}$$

This equation could be solved by the Newton method for a non-linear system of equations.

Although this method is precise enough, and even covers some well-known method in the literature, like [12] and [13], it is quite time-consuming due to the iterations. Fig. 3 illustrate the iterative approach leading to the approximated foot-point. In each iteration, the point moves to a lower level curve till reaching the zero level curve. Simultaneously, the gradient direction at each iteration is adapted to be parallel to the connecting line.

Instead of computing the real shortest distance, [9] proposes some estimation avoiding iteration. For the correspondence problem they proposed to restrict the search along the estimated normal direction. The estimated normal at each data point is computed by using principal components analysis (PCA). Precisely speaking, the covariance matrix of a set of points in the neighborhood is computed at first, and then its smallest eigenvector[2] is adopted as the orientation. (Fig. 2(b) shows an illustration in 2D space).

After finding the correspondence set for the given data set, the surface parameters should be refined through some minimization. However, on the contrary to algebraic approaches where least square method gives a unique and direct solution, geometric approaches need a different framework for finding the optimal set

[2] Eigenvector associated with the smallest eigenvalue.

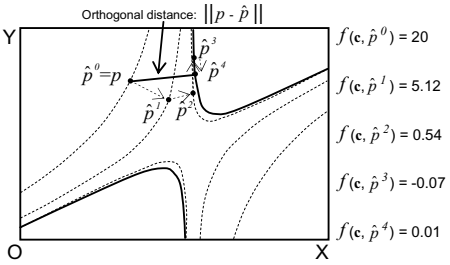


Fig. 3. Orthogonal distance computed by means of the iterative approach proposed in [7]. Solid curve correspond to the $f(\mathbf{c}, \mathbf{X}) = 0$, while dotted ones show the level curves obtained after each iteration of (5); $\hat{p}$ converges to the curve after four iterations.

of parameters. Next section presents one of the most well-known optimization frameworks adopted by most of geometric approaches.

Surface parameters refinement: As a result from the previous stage the set of points $\{\hat{p}_i\}_{i=1}^n$, corresponding to every p_i in $\mathbf{X}$ has been found. Hence, now an optimization framework is used to refine the surface parameter. LevenbergMarquardt algorithm (LMA) is a well-known method in non-linear optimization [14], which in some sense interpolates between the GaussNewton algorithm and the gradient descent.

In order to handle LMA, the value of the functional (inner part of (4)) and its partial derivatives, which is expressed in the Jacobian matrix, should be provided. Since each $\hat{p}_i$ lies on the surface, every distance $d(p_i, \hat{p}_i)$ can be easily expressed as a function of surface parameters:

$$d_i(\mathbf{c}) = \|p_i - \hat{p}_i\|^2, \tag{6}$$

more precisely, equation $f(\mathbf{c}, \hat{p}_i) = 0$ provides a link between surface parameters and $\{\hat{p}_i\}_{i=1}^n$ set, and distances as a consequence. This relationship is used to compute the Jacobian matrix:

$$J_{ij} = \frac{\partial d_i}{\partial c_j} = -\frac{\partial d_i}{\partial \hat{p}_i} \frac{\partial f / \partial c_j}{\partial f / \partial \hat{p}_i}. \tag{7}$$

Having estimated the geometric distance (6) and its Jacobian matrix (7), it is easy to refine the surface parameter through LMA as follows:

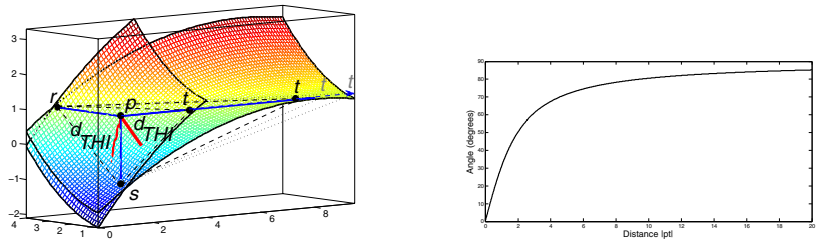
$$\begin{aligned} \boldsymbol{c}^{t+1} &= \boldsymbol{c}^t + \beta \triangle \boldsymbol{c}, \\ (J^T J + \lambda diag(J^T J)) \triangle \boldsymbol{c} &= J^T D, \end{aligned} \tag{8}$$

where β is the refinement step; $\triangle \mathbf{c}$ represents the refinement vector for the surface parameters; λ is the damping parameter in LMA; and the vector $D = (d_1(\mathbf{c}^t), ..., d_n(\mathbf{c}^t))^T$ corresponds to the distances. Parameter refinement (8) must be repeated till convergence is reached.

3 Proposed Approach

So far, the geometric approaches have been concerned about the best *direction* (i.e., the shortest distance) in each data point toward the surface. Taubin [12] approximate this direction with the gradient vector of the level set passing through the data point. Ahn et al. [7] present different optimization models in order to reach a better approximation of the best direction. They show that the method used by Taubin is a special case of their general method.

In the current work, a novel estimation of the geometric distance is presented, which despite other approaches, is not based on a single direction. First a tetrahedron is constructed, and then the geometric distance is approximated with the tetrahedron height segment; this tetrahedron is easily defined by the given point and three intersections satisfying $f_{\mathbf{c}}(x, y_p, z_p) = 0$, $f_{\mathbf{c}}(x_p, y, z_p) = 0$ and $f_{\mathbf{c}}(x_p, y_p, z) = 0$, where (x_p, y_p, z_p) is the given point. Fig. 2(c) shows an illustration for the 2D case; the 3D case in depicted in Fig. 4(a).



(a) *(b)*

Fig. 4. Transition from a tetrahedron height vector to a triangle height vector. (a) Illustration of two surfaces with their corresponding intersection points (planar triangular patch (r, s, t)). (b) Angle between tetrahedron height segment and (pt) segment, as a function of distance $|pt|$.

In the particular case tackled in this work, since the fitted surface is defined by an implicit quadric equation $f_{\mathbf{c}}(x, y, z) = 0$, the intersection points can be easily found by solving some quadratic equations. Each equation gives two roots, the closest one to the data point is selected as the vertex of the tetrahedron.

A direct formula to describe the proposed distance can be found. Let r, s and t be the three intersections with a quadric surface, which create a triangular planar patch (see Fig. 4(a)). Since the volume of the tetrahedron is defined as the product of the area of each base by its corresponding height, three pairs of expressions lead us to the same value. Hence, the height of the tetrahedron, d_{TH}, could easily be computed from the following relationship:

$$(|\boldsymbol{rs} \times \boldsymbol{rt}|.|d_{TH}|)/6 = (|pr|.|ps|.|pt|)/6 = \upsilon, \tag{9}$$

where $\times$ refers to the cross product operator between two vectors. Similar relationship can be found in the 2D case, but by using the area of the triangle instead of the volume.

Note that in the extreme cases, when intersections with some of the directions $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ cannot be found, the 3D case becomes into: *i*) the 2D case (two intersections); *ii*) only one intersection, which has been used in [8]; or *iii*) the point p is an outlier since none of the three directions intersects the implicit quadric surface. Transitions between different cases are smoothly reached; Fig. 4(*a*) shows an illustration where one of the vertices of the triangular patch (r, s, t) moves away from current position up to the extreme—i.e., no intersection between vertex t and the implicit surface can be found[3]; the smooth transition from the tetrahedron height segment orientation to the triangle height segment orientation can be appreciated in the illustration of Fig. 4(*b*).

This proposed measurement criterion can be exploited in a fitting framework. Indeed, the function $\sum d_i^2$ could be a good optimization model for the surface fitting, where each d_i is the proposed distance, d_{TH}, for the point p_i in the data set. Based on (9), this function has a nonlinear relationship with the surface parameters. Hence, for this part a nonlinear optimization method is needed to find the best set of parameters.

In the current work, the LMA is adopted, which has been presented in section 2.2. As mentioned above, the Jacobian matrix, which shows the sensitivity of each d_i with respect to the parameter vector, needs to be computed. For this purpose (9) could be used to describe d_{TH} based on the surface parameters, and consequently we have:

$$D_j|d_{TH}| = (|\boldsymbol{rs} \times \boldsymbol{rt}|.D_j\upsilon - \upsilon.D_j|\boldsymbol{rs} \times \boldsymbol{rt}|)/|\boldsymbol{rs} \times \boldsymbol{rt}|^2, \tag{10}$$

where $D_j = \partial/\partial c_j$. All these terms are based on the coordinates of the intersections, and since every intersection r, s, and t is implicitly related to the parameter vector, the derivations could be easily computed like (7).

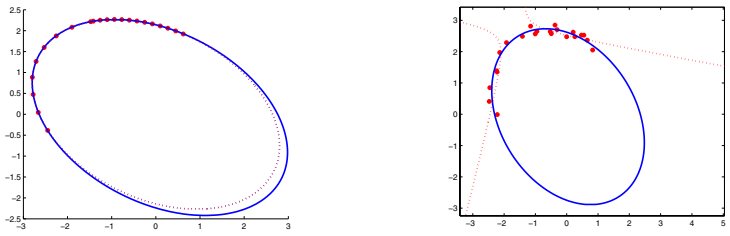
4 Experimental Results

Two major approaches, the *algebraic* and *geometric* one, with their extensions have been presented. Algebraic methods are quite fast, but unfortunately the structure of the data is neglected. On the other hand, geometric methods are more afraid of the geometry of the data, as the name implies, but they are quite slow. The proposed method, which belongs to the geometric category, is implemented and compared with the most important methods in the literature.

In the two dimensional case, a set of points picked from an ellipse with a non-uniform distribution is used. Fig. 1(b) presents the result from the 3L algorithm [6], which fails to fit the right ellipse, even though the case is not noisy. However, the simple algebraic (3) manages to fit it, but the point is that the 3L algorithm is supposed to include more geometric information than the least square approach.

Fig. 5(a) depicts the result of the proposed method for the same set of points. Both algebraic and proposed method converge to a similar result, but problems

[3] These experiments have been performed by applying smooth changes in the geometry of the surface, which correspond to smooth changes in the parameter space.



(a) *(b)*

Fig. 5. Fitting a 2D set of points picked non-uniformly from an ellipse. (*a*) Without noise: Algebraic (dotted line) and proposed method (solid line) both reach similar results. (*b*) Noisy data case: Algebraic method (dotted line) misses the elliptic structure, while the proposed approach (solid line) reaches a good result.

arise when some noise is added to the points. Fig. 5(b) highlights the robustness of the proposed method to noise; whereas the algebraic one missed the elliptic structure of the data, and fitted the patch as a split hyperbola. Fitzgibbon et. al. [15] proposed 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.

The proposed approach has been also compared with the state of the art by using real range images obtained with the the K2T structured light camera system, University of South Florida. A first data set, from a spherical object, is presented in Fig. 1(a). It contains 1000 points, and as indicated in that figure, the algebraic method fails to fit the right sphere; since this approach is just trying to put the *algebraic* value of the quadratic function closer to *zero*, a wrong result is achieved. Fig. 6 shows the result obtained by the proposed method, when the same set of points is considered. In this case a sphere with a bigger radius covering the whole data set has been used as an initialization. The whole process took 4.26 sec. in a 3.2GHz Pentium IV PC with a non-optimized Matlab code. The other geometric method, proposed by [7], reaches similar result but 10 times slower. Since at each parameter refinement iteration there are additional iterations to find the foot-points.

Fig. 7 shows a 3D patch picked from a cylinder corrupted by a Gaussian noise. This patch was generated and rotated in a synthetic way. Three different methods are compared based on the accumulated real distances from the points to the achieved surface, computed by using [7]. Fig. 7(a) and (c), respectively, show the results of the 3L algorithm [6] and the proposed method. Both images show similar result, one concave and the other one convex though. The first one has an accumulated real distance of 28.90 and the proposed one has an accumulated distance of 5.45, which is more than five times smaller. Fig. 7(b) shows the result from another geometric approach [7]. This result has the lowest distance (5.24), and it is still a cylinder, but it took more than five times compared with the proposed algorithm.

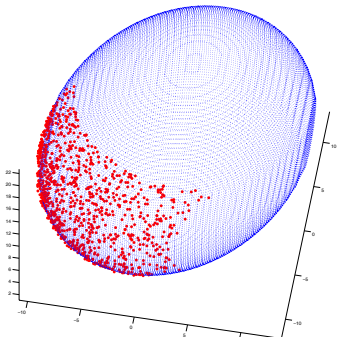
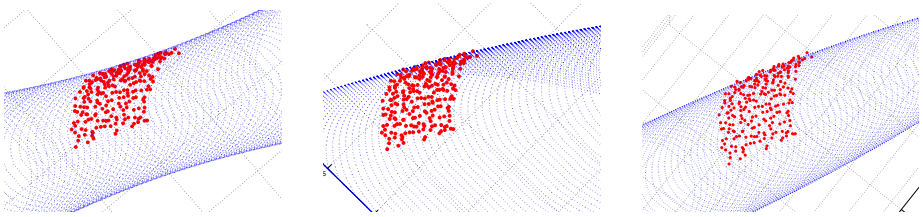


Fig. 6. Fitting real 3D data through the proposed approach



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

Fig. 7. Fitting of the 3D patch of a cylinder disrupted by gaussian noise for: (*a*) the 3L algorithm[6]; (*b*) the geometric method proposed by [7]; (*c*) the proposed method

In the last example another comparison is presented. This time a part of a noisy ellipsoid is used as an input of both algebraic and geometric approaches. Fig. 8(a) illustrates how the simple algebraic method (3) misses the elliptic structure of the patch, and gives a hyperboloid. This example shows the importance of using geometric clues. Fig. 8 (b) and (c) illustrate the 3L results, with different parameters. As mentioned in section 2.1, the 3L algorithm, as the name stands for, needs two other data sets that are offsets of the original one. Here, the offsets are constructed with different user defined parameters, which show the amounts of translation. Fig. 8(c) shows the result of the 3L method with a smaller parameter; so as we continue to squeeze the offset, the result makes more sense, but unfortunately after a while on, since the data is noisy, three level sets merged together and the whole procedure collapses. Finally, Fig. 8(d)) depicts the result of the proposed method, which manages to fit the data and maintain the elliptic curvature as well. It should be mentioned that, the proposed distance estimation does not need any parameter adjustment. The result from [7] is skipped, because it obtains quite the same result but with a slower convergence.

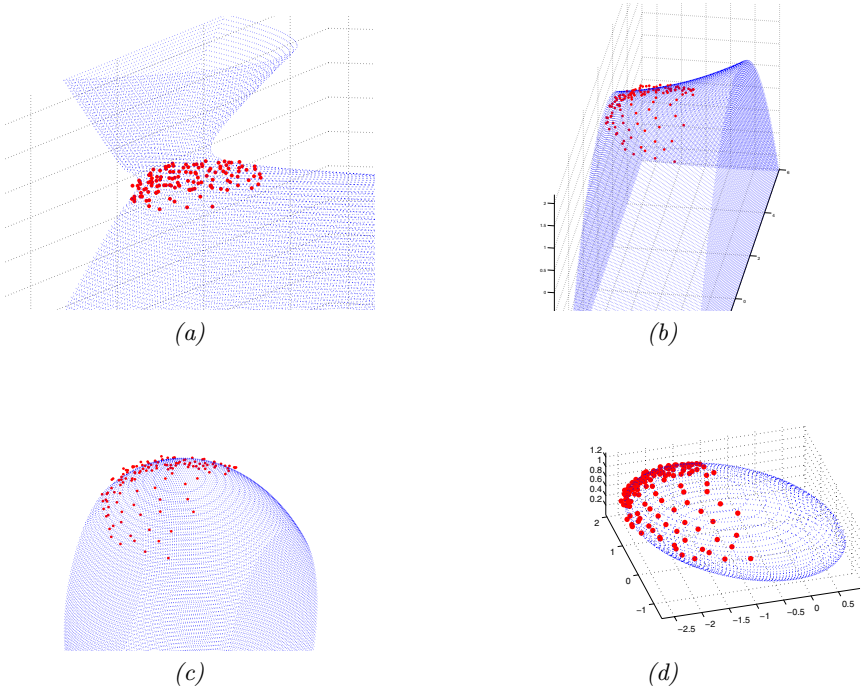


Fig. 8. Fitting 3D points picked from an ellipsoid with a non-uniform distribution. Final results from: (*a*) the simple algebraic method (3); (*b*) and (*c*) the results of the 3L algorithm [6] with different set parameters; and (*d*) proposed method.

5 Conclusions

This paper presents a throughout study of state of the art fitting methods. Furthermore, a novel geometric distance estimation is proposed. Despite other geometric estimations, which are based on one direction to find the foot-point associated to each data point, the proposed one is based on three different directions; hence it does not need any iterations. This approximation value has been used with a LMA optimization framework. In that framework an iterative approach finds the best set of surface parameters. Compared with other geometric methods, instead of relying on a costly iterative approach to find the foot-points, a direct way is proposed. As a conclusion, it can be said that even though several algorithms have been proposed for quadric fitting there is a trade off between CPU time and accuracy of surface parameters for selecting the best one; this trade off get more evident when it is increased the number of points to be fitted or the percentage of noise. The only concern arises in the proposed method when no intersection could be found for some data points; however, the proposed method is flexible enough to apply there other direction instead of the coordinate axis. Future work will study this possibility.

References

1. Besl, P., Jain, R.: Segmentation through variable-order surface fitting. IEEE Trans. on Pattern Analysis and Machine Intelligence 10(2), 167–192 (1988)
2. Aigner, M., Jutler, B.: Gauss-newton-type technique for robustly fitting implicit defined curves and surfaces to unorganized data points. In: IEEE International Conference on Shape Modelling and Application, pp. 121–130 (2008)
3. Yang, Z., Deng, J., Chen, F.: Fitting unorganized point clouds with active implicit b-spline curves. Visual Computing 21(1), 831–839 (2005)
4. Carr, J., et al.: Reconstruction and representation of 3d objects with radial basis functions. SIGGRAPH, 67–76 (2001)
5. Helzer, A., Barzohar, M., Malah, D.: Stable fitting of 2d curves and 3d surfaces by implicit polynomials. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(10), 1283–1294 (2004)
6. Blane, M., Lei, Z., Civil, H., Cooper, D.: The 3l algorithm for fitting implicit polynomials curves and surface to data. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(3), 298–313 (2000)
7. Ahn, S., Rauh, W., Cho, H., Warnecke, H.: Orthogonal distance fitting of implicit curves and surfaces. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(5), 620–638 (2002)
8. Chen, C., Hung, Y., Cheng, J.: Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(11), 1229–1234 (1999)
9. Gotardo, P., Bellon, O., Boyer, K., Silva, L.: Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics 34(6), 2303–2316 (2004)
10. Keren, D., Gotsman, C.: Fitting curves and surfaces with constrained implicit polynomials. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(1), 476–480 (1999)
11. Tasdizen, T., Tarel, J., Cooper, D.: Improving the stability of algebraic curves for applications. IEEE Trans. Image Processing 9(3), 405–416 (2000)
12. Taubin, G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 13(11), 1115–1138 (1991)
13. Sampson, P.D.: Fitting conic sections to very scattereda data: An iterative refinement of the bookstein algorithm. Computer Graphics and Image Processing, 18, 97–108 (1982)
14. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (1990)
15. Fitzgibbon, A., Pilu, M., Fisher, R.: Direct least square fitting of ellipses. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 476–480 (1999)