

# Relaxing the 3L algorithm for an accurate implicit polynomial fitting

Mohammad Rouhani  
Computer Vision Center  
Edifici O, Campus UAB  
08193 Bellaterra, Barcelona, Spain  
rouhani@cvc.uab.es

Angel D. Sappa  
Computer Vision Center  
Edifici O, Campus UAB  
08193 Bellaterra, Barcelona, Spain  
asappa@cvc.uab.es

## Abstract

*This paper presents a novel method to increase the accuracy of linear fitting of implicit polynomials. The proposed method is based on the 3L algorithm philosophy. The novelty lies on the relaxation of the additional constraints, already imposed by the 3L algorithm. Hence, the accuracy of the final solution is increased due to the proper adjustment of the expected values in the aforementioned additional constraints. Although iterative, the proposed approach solves the fitting problem within a linear framework, which is independent of the threshold tuning. Experimental results, both in 2D and 3D, showing improvements in the accuracy of the fitting are presented. Comparisons with both state of the art algorithms and a geometric based one (non-linear fitting), which is used as a ground truth, are provided.*

## 1. Introduction

Implicit polynomials (IPs) have shown to be useful in different computer vision applications. Their capability to describe complicated boundaries through their coefficient vector, and the nonexistence of parametrization, has been exploited in fields such as: pose estimation [6], [12], shape description [7], position invariant object recognition [8], 3D image segmentation [14] and registration [13], to mention a few.

According to the metric used for measuring the accuracy of the approximation, classical approaches of IPs fitting can be divided into two categories: (i) *orthogonal* or *geometric distance* based (e.g. [1], [2], [4], [11]) and (ii) *algebraic distance* based (e.g. [3], [5]), each one of them has its own advantage and disadvantage. In short, orthogonal distance based approaches reach more accurate results while algebraic ones are more efficient computationally. Although the current work is focussed on this second category the accuracy of the obtained result can be compared with the orthogonal based approaches.

This paper has two main contributions. On the one hand, the proposed framework allows to relax the additional constraints of the 3L so that a more accurate result can be reached even using such a kind of algebraic approach. On the other hand, it does not require an accurate initial guess for the thresholds needed by the 3L algorithm. The rest of the paper is organized as follows. Section 2 describes the problem of fitting implicit polynomials and briefly introduces the 3L algorithm. The proposed technique is presented in section 3. Section 4 gives experimental results and comparisons.

## 2. Problem formulation and background

### 2.1. Implicit polynomial fitting

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  $f(\mathbf{x})$  be an implicit polynomial of degree  $d$  represented as:

$$f(\mathbf{x}) = \sum_{\substack{(i+j+k) \leq d \\ \{i,j,k\} \geq 0}} a_{i,j,k} \cdot x^i \cdot y^j \cdot z^k = 0, \quad (1)$$

or, in a vector form:

$$f(\mathbf{x}) = \mathbf{m}^T \mathbf{a} = 0, \quad (2)$$

where  $\mathbf{m}$  is the column vector of monomials and  $\mathbf{a}$  is the polynomial coefficient vector; the fitting problem consists in first defining a criterion—or residual error—to measure the closeness of the zero set to the given data set, and then minimizing this criterion to find the best coefficient vector  $\mathbf{a}$ . Let  $\Gamma_0$  be the set of given data points with coordinates  $\mathbf{x}$  (picked up from object boundaries in 2D or surfaces in 3D). Since IPs are chosen to describe the data, their value deviation from zero in each point could be considered as a residual error:

$$E = \sum_{\mathbf{x} \in \Gamma_0} f^2(\mathbf{x}), \quad (3)$$

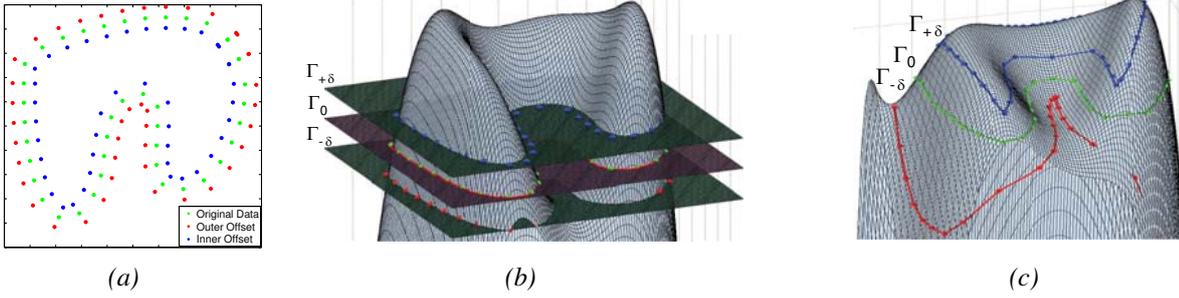


Figure 1. (a) Level sets: original data ( $\Gamma_0$ ), outer offset ( $\Gamma_{-\delta}$ ) and inner offset ( $\Gamma_{+\delta}$ ). (b) A 3D illustration of the original 3L algorithm. (c) The best fitting polynomial obtained by [1], showing that the values in the inner and outer sets should be relaxed.

or, in vector form:

$$E = \mathbf{a}^T \mathbf{M}_{\Gamma_0}^T \mathbf{M}_{\Gamma_0} \mathbf{a}. \quad (4)$$

where  $\mathbf{M}_{\Gamma_0}$  is the matrix containing monomial vectors calculated in each data point. This residual error is referred in the literature as algebraic distance.

## 2.2. Overview of the 3L algorithm

The simplest fitting approach is the Least Squares method (LS), which minimizes the algebraic distance presented above. Unfortunately, there is not a clear geometric interpretation of (4) since it does not measure the orthogonal distance from the data points to the polynomial zero set. Furthermore, solutions obtained by the least squares method, although faster, are very unstable.

In order to compensate the lack of geometric meaning, and to solve the instability problem in the classical algebraic methods (4), the authors in [3] have proposed the 3L algorithm, which consists in generating two additional *level sets*:  $\Gamma_{-\delta}$  and  $\Gamma_{+\delta}$  from the original data set  $\Gamma_0$ . These two additional data sets are generated so that one is internal and the other is external. These sets are placed at a distance  $\pm\delta$  from the original data along a direction that is locally perpendicular to the given data set. In the current implementation a PCA based approach, in a local neighborhood for every point, has been used for estimating this direction. Hence, the 3L algorithm incorporates a control for a local continuity resulting in a more stable solution. Figure 1(a) depicts the original data with the two additional level sets; the corresponding 3D illustration is presented in Fig. 1(b).

The 3L fitting algorithm is then formalized as a linear least squares *explicit* polynomial fitting problem. Considering the three level sets:  $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$  the equation (2) is now defined by using a block matrix  $\mathbf{M}_{3L}$  and a block column vector  $\mathbf{b}$ :

$$\mathbf{M}_{3L} = \begin{bmatrix} \mathbf{M}_{\Gamma_{-\delta}} \\ \mathbf{M}_{\Gamma_0} \\ \mathbf{M}_{\Gamma_{+\delta}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -c \\ \mathbf{0} \\ +c \end{bmatrix}, \quad (5)$$

where  $\mathbf{M}_{\Gamma_0}$ ,  $\mathbf{M}_{\Gamma_{+\delta}}$ ,  $\mathbf{M}_{\Gamma_{-\delta}}$  are matrices of monomials calculated in the original, inner and outer set respectively;  $\pm c$

are the corresponding expected values in the inner and outer level sets. Then, the least squares solution for  $\mathbf{a}$  is obtained:

$$\mathbf{a} = \mathbf{M}_{3L}^\dagger \mathbf{b} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L})^{-1} \mathbf{M}_{3L}^T \mathbf{b}, \quad (6)$$

where  $\mathbf{M}_{3L}^\dagger$  denotes the pseudoinverse of  $\mathbf{M}_{3L}$ .

A crucial aspect of the 3L algorithm is the generation of the two additional level sets. According to [3], the value of  $\delta$  could be specified as a percentage of the object size. However there is not a general rule for tuning this value; it depends on the nature of the problem (*i.e.*, shape of the contour/surface, density of points, degree of the IP to be fitted). Figure 2 illustrates different results obtained with the 3L algorithm by setting  $\delta$  as a: 1, 3, 6, 10, and 25 percent of the object size. It should be mentioned that the best fitting corresponds to  $\delta = 4$  percent of the object size, see Fig. 8(a). Through the whole paper the accumulated fitting error, hereinafter AFE, is used to measure the accuracy of the obtained result. The AFE is computed over the original data set and it corresponds to the shortest distance between each point and the fitted IP (usually referred in the literature as Euclidean or orthogonal distance).

Other algebraic works have been proposed in order to increase the stability of the least square solution. In [10] the authors exploit geometric information of the data related to their orientation. In other words, they approximate the normal orientation of the data set in each point, and try to find a polynomial having gradient vectors similar to these normal vectors. To define the problem in a single unified least squares form, they force the gradients to have a unit length; the algorithm is referred as Gradient-One algorithm. This method has been later improved in [5] in order to have a more stable solution. Firstly, the stability of the zero set with respect to small changes in the coefficient vector is formulated. Then, two different approaches, named Min-Max and Min-Var, are proposed to find the solution. In the first approach an upper bound of the error function is minimized, while in the second one the minimization is focussed on the variance of that error function.

The ridge regression technique has been applied in [9]

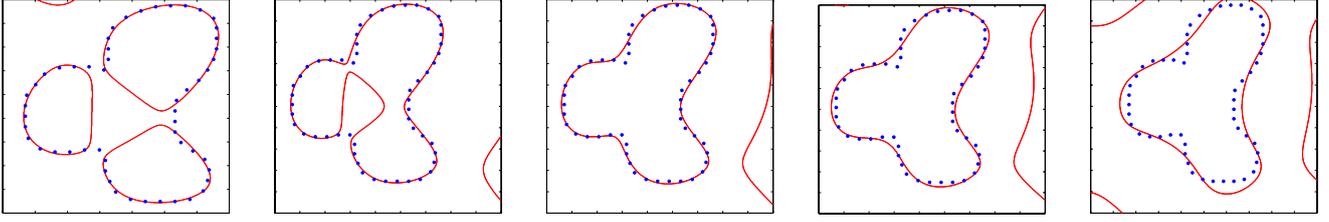


Figure 2. Fifth degree fitting results obtained with the 3L algorithm by setting  $\delta$  as a 1, 3, 6, 10, and 25 percent of the object size (the best results correspond to  $\delta = 4$  percent of the object size, see Fig. 8(a)).

to tackle the instability problem of least squares in (6); it regularizes the least squares formulation through adding a diagonal matrix to the monomial matrix. The authors of that work propose some conditions on the diagonal matrix to maintain the Euclidean invariance while regularizing the LS solution.

Regardless to the pros and cons of these improvements, none of them pay attention to the expected values of the additional data in (5). Indeed, forcing the IP reaching  $+c$  inside and  $-c$  outside could lead us to a biased result. However, changing the value of  $c$  will give the same zero set. What makes the fitting result biased is the constant proportion considered for all the inner and outer additional data points. Figure 1(c) depicts an illustration where the perfect result<sup>1</sup> would only be obtained after relaxing the expected values for the inner and outer offsets, while keeping them to zero for original data. This figure shows the original data and the expected height for the supplementary offsets; and it also illustrates how these values change from point to point. The main contribution of the current work is to estimate these expected values in order to reach a more accurate fitting.

### 3. Proposed approach

As mentioned above, the 3L algorithm is based on the construction of two additional data sets (level sets) that are determined from the original data set. Although the algorithm produces a result within one pass and no iterative computations are required, it has three major problems as detailed below.

First of all, like other algebraic methods, the 3L algorithm has also numerical instability problems. In a recent work, [9] proposes a statistical approach to increase the global stability of the 3L algorithm. This method is based on the ridge regression (RR), which is a way to regularize the block matrix  $\mathbf{M}_{3L}$ . RR improves the condition number of the block matrix by adding a diagonal matrix  $\mathbf{D}$  to  $\mathbf{M}_{3L}^T \mathbf{M}_{3L}$ :

$$\mathbf{a} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L} + \kappa \mathbf{D})^{-1} \mathbf{M}_{3L}^T \mathbf{b}. \quad (7)$$

Moreover, in [9] a diagonal matrix is introduced such that the Euclidean invariant property is maintained. Unfortunately, this method leads us to a coarser fitting result than the original 3L algorithm, although more stable.

Second, the accuracy of the fitting result depends on the  $\delta$  value used for computing the two additional level sets. In the original paper [3] the authors propose to define these offsets as a percentage of the object size; however there is no rule for setting the right value. Actually, in [3] the authors present some experiments showing the variability of the results depending on that value (varying these offsets from 0.5 percent till 20 percent of the object size).

Third, it should be noticed that the whole set of points contained in the three level sets  $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$  defining the block matrix  $\mathbf{M}_{3L}$  are equally considered when the least squares solution is obtained (6). On the other hand, the initially given problem only contains the level set  $\Gamma_0$ . Hence, it is easy to conclude that the constraints added for stabilizing the fitting solution could affect the accuracy of the result.

In the current work the last two problems mentioned above are tackled looking for a more accurate result. A strategy for relaxing the additional constraints is proposed to decrease the fitting error while maintaining the structural shape of the object. The proposed approach consists of two stages as detailed below.

#### 3.1. Relaxing additional constraints

As mentioned earlier, the 3L algorithm tries to encode the geometric information of the data by adding two supplementary sets supporting the original one. In the original work [3], it is suggested to have an equal value for the whole inner and outer offsets. However, as shown in Fig. 1(c), the perfect fitting result could be only reached by means of different values for each point in the inner and outer data sets. In other words, the right hand values in (5) (constant vectors  $\pm c$ ) should be relaxed in order to obtain the most accurate fitting result. In the current work a novel idea to adjust these values for offsets, based on the position of the point and the approximated IP at the current iteration, is proposed.

<sup>1</sup>This result corresponds to the fitting of the given data points computed with [1] (see Section 4), inner and outer data sets are not considered.

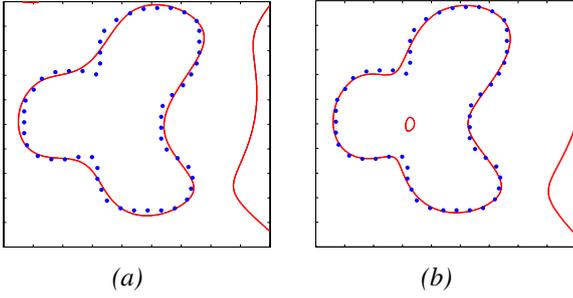


Figure 3. (a) Coarse approximation used as an initialization of the proposed approach ( $\delta = 10$  percent of the object size, AFE=0.1989). (b) Fitting result obtained with the proposed approach (four iterations, AFE=0.1288).

Lets  $f(\mathbf{x})$  be the IP at the current iteration;  $p_i = (x_i, y_i)$  a given data point<sup>2</sup>;  $s_i$  and  $t_i$  its inner and outer offset respectively. As mentioned above these two points are obtained along the unit normal  $\mathbf{n}_i = (n_i^x, n_i^y)$  from the local PCA based approximation. Moving on the surface from  $s_i$  to  $t_i$  can be parameterized as follow:

$$g(t) = f(x_i + n_i^x t, y_i + n_i^y t), \quad (8)$$

where  $f$  shows the value of the fitted IP (see (2)). The expected value for  $g(0) = f(x_i, y_i)$  is zero, but its value in  $s_i$  and  $t_i$  must be estimated.

Based on the above definition,  $g(\delta)$  and  $g(-\delta)$  show the value of the IP achieved in  $s_i$  and  $t_i$  respectively. Considering the function  $g$  at these two points a fair proportion for next iteration could be obtained, instead of using the fixed values  $\pm c$  in (5). For this purpose a first order Taylor approximation, around  $t = 0$ , of (8) could be computed as:

$$g(\pm\delta) \approx [n_i^x f_x(x_i, y_i) + n_i^y f_y(x_i, y_i)]\delta. \quad (9)$$

However, this value could be approximated again by considering  $\mathbf{n}_i \approx \nabla f(p_i) = (f_x(x_i, y_i), f_y(x_i, y_i))^T$ :

$$g(\pm\delta) \approx \pm \|\nabla f(p_i)\|^2 \delta, \quad (10)$$

so the next expected value for the given point should be  $[0, g(\delta), g(-\delta)]$  for the original set and the inner and outer one respectively. This process is applied for every given point and then vector  $\mathbf{b}$  in (5) updated. Then, the least squares method (6) is used for computing the new IP coefficients. Note that this least squares solution is obtained just after a matrix multiplication, since the pseudoinverse matrix in (6) is computed only once, at the beginning. The whole procedure is iterated till convergence is reached as explained in the next section.

<sup>2</sup>Without loss of generality, and only to make clear its understanding, this discussion is presented for the 2D case, but it could be extended to the 3D case.

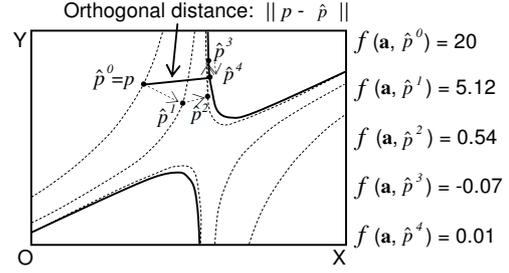


Figure 4. Orthogonal distance from a given point  $p$  computed by means of the iterative approach [1].

### 3.2. Convergence criteria

Stopping the above iteration represents a key point. On the one hand it should be something easier to compute; on the other hand it should be robust enough to be used with different geometries.

In the proposed method, a coarse fitting from the 3L algorithm is chosen as an initialization, and in each iteration the total angle between the gradient vector at each data and its approximated unit normal  $n_i$ , from local PCA (see Section 2.2), is measured. It should be mentioned that the approximated normal is already calculated when computing the two additional level sets. So the only required computation is regarding to the angle estimation:

$$\theta_i = \cos^{-1} \left( \frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|} \right), \quad (11)$$

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:

$$\xi(\mathbf{a}) = \sum_{i=1}^N 1 - \left| \frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|} \right|, \quad (12)$$

where  $N$  is the number of points in the original data set. The process iterates while (12) decreases.

Finally, it should be mentioned that in spite of the iterative nature of the proposed approach, it is more related to algebraic approaches than to the Euclidean based ones, where at every iteration the shortest distance between every single point and the current fitted IP should be computed [4]. Figure 3(a) shows the coarse fifth degree fitting from the 3L algorithm used as an initialization of the proposed approach; the final fitting result is obtained after four iterations (Fig. 3(b)).

## 4. Experimental results

This section presents experimental results obtained with the proposed approach as well as comparisons with both the state of the art algorithms and the original 3L formulation. Additionally, an orthogonal distance based approach

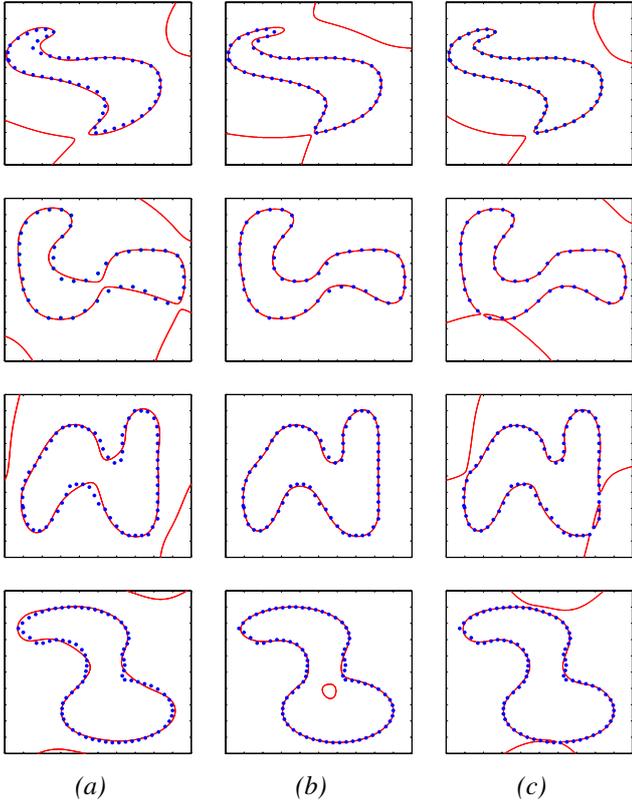


Figure 5. (a) Results from the 3L algorithm. (b) Results from the proposed approach. (c) Results from [1] used as ground truths.

has been implemented in order to be used as a ground truth; it is briefly introduced next.

**Ground truth.** In [1] an orthogonal distance based fitting is proposed, which for every single point  $p$  searches the corresponding foot-point,  $\hat{p}$ , on the surface satisfying  $f(\hat{p}) = 0$ . Furthermore, the line connecting the data point with the foot-point must be parallel to the  $\nabla f$  at the foot-point, where  $\nabla$  is the gradient operator. In other words, we must have  $\nabla f \times (\hat{\mathbf{p}} - \mathbf{p}) = 0$ . Merging these two conditions, the following system of equations must be solved:

$$\begin{pmatrix} f \\ \nabla f \times (\hat{\mathbf{p}} - \mathbf{p}) \end{pmatrix} = \mathbf{0}. \quad (13)$$

This equation could be solved by the Newton method for non-linear system of equation. Figure 4 shows an illustration of this iterative approach leading to the approximated foot-point for a given 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. Once the foot-point for every single point is found the total square distance is minimized through a non-linear method (e.g., Levenberg-Marquardt).

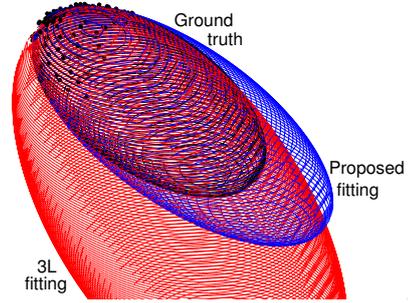


Figure 6. Non-uniform synthetic noisy data from an ellipsoid (127 points), fitted with the 3L algorithm ( $\delta = 10\%$ , AFE=4.3376), the proposed approach (AFE=3.1234) and [1] (AFE=3.2441).

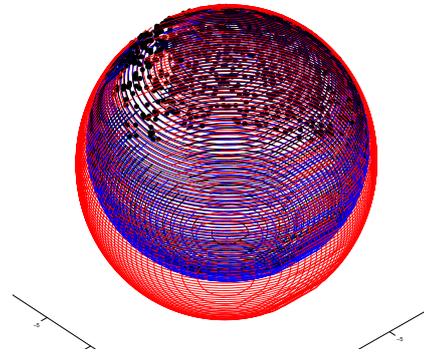


Figure 7. 3D data points obtained with a structured light camera from a sphere; outer mesh correspond to the result obtained with the 3L algorithm (AFE=242.6154) while inner spheres show the results from the proposed approach (AFE=85.1591) and the ground truth (AFE=85.4815).

This approach has been implemented just to be used as a ground truth and measure the accuracy of the results obtained with the proposed approach.

**2D and 3D data sets.** Several data sets have been fitted with the proposed approach and compared with the best results obtained with the 3L algorithm. Furthermore, the results obtained with [1] are provided. Figure 5 shows 2D contours fitted by sixth degree IPs using the 3L algorithm (Fig. 5(a)), the proposed approach (Fig. 5(b)) and a non-linear orthogonal distance based approach [1] (Fig. 5(c)). In all the cases the accuracy obtained with the proposed approach improves considerable the one obtained with the 3L algorithm (see Table 1); moreover, it is comparable 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 the proposed approach is about ten times faster than [1].

Figure 6 depicts the fitting results obtained with the three approaches when 3D data points are considered; note that although similar AFEs are obtained the geometry computed with proposed approach is more similar to the ground truth

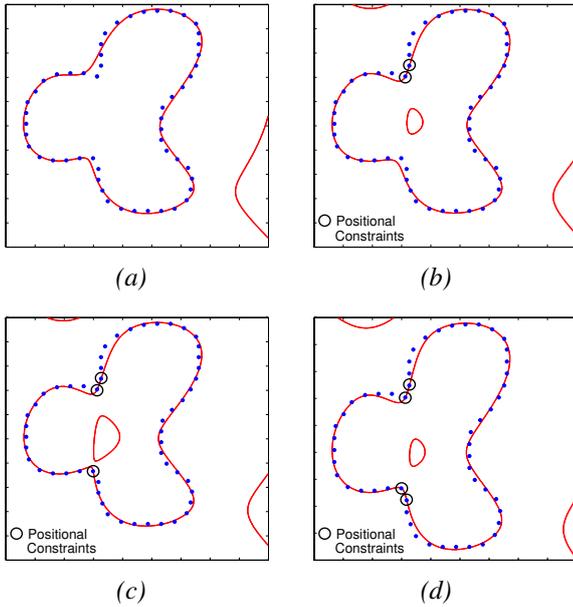


Figure 8. (a) Best result from the 3L algorithm obtained by setting  $\delta = 4$  percent of the object size (AFE=0.1285). Results after incorporating: (b) two positional constraints (AFE=0.1179); (c) three positional constraints (AFE=0.1068); and (d) four positional constraints (AFE=0.1105).

than the one obtained with the 3L. Finally, Fig. 7 presents results obtained after fitting a set of real data points corresponding to a partial view of a sphere. In this case not only qualitative better results are obtained with the proposed approach but also quantitatively.

**Zero set.** In order to force that the fitted IP pass through particular data points, known as positional control points, [3] proposes to incorporate additional linear constraints. Figure 8(a) shows the best fitting result obtained with the 3L algorithm. It has been achieved after trying different  $\delta$  values and it corresponds to a  $\delta = 4$  percent of the object size. Figure 8(b), (c) and (d) depict results from the 3L algorithm after incorporating additional constraints, as proposed in [3], to force that two, three and four points respectively belong to the zero set. It can be concluded that increasing the number of positional constraint does not result in a reduction of the AFE. The result of the proposed approach (see Fig. 3(b))(AFE=0.1288) is quite similar to the values obtained after manually tuning  $\delta$  in the 3L algorithm or after adding positional control points.

**Non-uniform sampling and open boundaries.** Two particular challenges for fitting algorithms arise when data points are non-uniformly distributed or when they correspond to an open contour/surface. Figure 9 presents two illustrations obtained when the 3L algorithm, the proposed approach and [1] were used. Note that the results obtained with the proposed approach (AFE) are quite similar to the

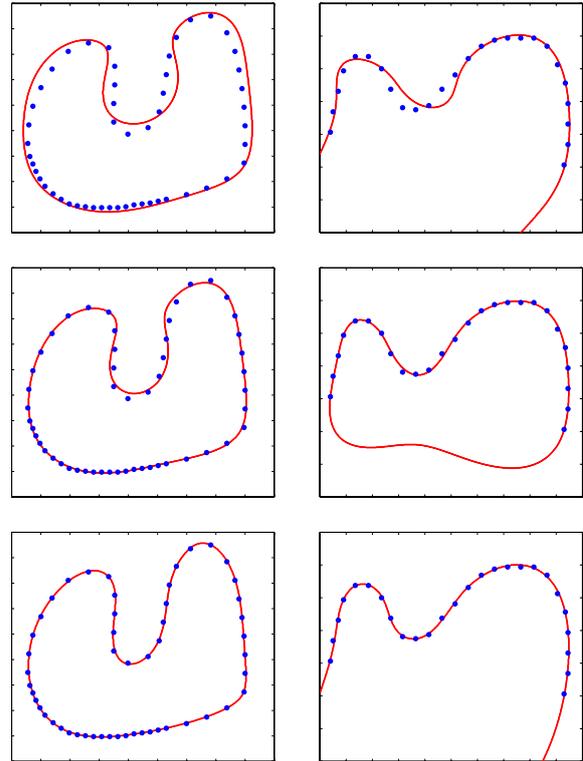


Figure 9. Non-uniform sampling and open boundary cases: (top) Results from the 3L algorithm; (middle) Results from the proposed approach; (bottom) Results from an orthogonal distance based approach [1], non-linear fitting.

ones obtained by using [1] (see Table 1), but they were generated almost ten times faster. A fourth degree IP was fitted in the open contour case (Fig. 9(right)) while a fifth degree IP was considered for the non-uniform point distribution case (Fig. 9(left)).

**Quantitative Comparisons.** Table 1 shows the AFE for five different methods as well as the proposed one. It should be noticed that the last column corresponds to [1] that is obtained by solving a nonlinear optimization, and needs more computation. As mentioned above it is used as a ground truth. All other methods belong to the algebraic category, which are solved by a simple least squares method. The proposed algorithm has obtained similar results to the geometric one, while is much faster. The Min-Max and Min-Var methods [5] are similar to the Gradient-One algorithm [10], which incorporates orientation in the optimization framework. All these methods try to obtain more *stable* fitting results while neglecting the accuracy. Fig. 10 shows how the Gradient-One algorithm fails to describe the corners. Even though the normal directions are preserved, but the zero set is away from the original data.

Table 1. Accumulated fitting errors to compare the results obtained by different approaches (3L: the 3L algorithm; GO: the Gradient One; MM: Min-Max; MV: Min-Var; PA: Proposed Approach; GA: Geometric Approach).

	3L[3]	GO[10]	MM[5]	MV[5]	PA	GA[1]
Fig. 5-(1st row)	5.70	13.24	13.76	13.75	<b>1.59</b>	0.84
Fig. 5-(2nd row)	4.82	13.96	3.91	10.48	<b>1.65</b>	0.96
Fig. 5-(3rd row)	8.36	16.50	10.96	15.82	<b>4.11</b>	3.75
Fig. 5-(4th row)	7.98	10.04	5.91	8.04	<b>2.74</b>	3.95
Fig. 9-(left col)	13.70	30.13	8.76	22.71	<b>1.22</b>	0.55
Fig. 9-(right col)	3.40	10.33	5.47	8.80	<b>0.73</b>	0.84

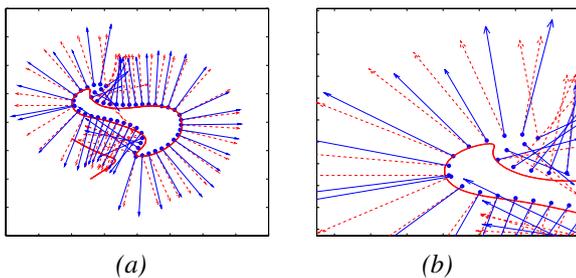


Figure 10. (a) The result obtained by the Gradient-One algorithm [10]. (b) An enlargement showing how the fitting algorithm ignores the positional constraint, the gradient vectors have a similar length and orientation though.

## 5. Conclusions

This paper presents a method for relaxing additional constraints of the 3L algorithm. In this way the accuracy of the fitting is increased as well as there is no need for a fine tuning of the two additional level sets. Experimental results fitting different IPs are provided showing the advantages of the proposed approach. Comparisons with the 3L and state of the art algorithms, as well as with a non-linear orthogonal distance based approach used as a ground truth, are given.

## Acknowledgments

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-2008ITT00001.

## References

[1] 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*, 24(5):620–638, May 2002. 1, 2, 3, 4, 5, 6, 7

[2] M. Aigner and B. Jutler. Gauss-newton-type technique for robustly fitting implicit defined curves and surfaces to unorganized data points. *IEEE International Conference on Shape Modelling and Application*, pages p 121–130, 2008. 1

[3] 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*, 22(3):p 298–313, March 2000. 1, 2, 3, 6, 7

[4] P. Faber and R. Fisher. Pros and cons of euclidean fitting. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 414–420, London, UK, 2001. Springer-Verlag. 1, 4

[5] A. Helzer and M. Barzohar. Stable fitting of 2d curves and 3d surfaces by implicit polynomials. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1283–1294, 2004. Fellow-Malah, David. 1, 2, 6, 7

[6] G. Marola. A technique for finding the symmetry axes of implicit polynomial curves under perspective projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):465–470, 2005. 1

[7] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(8):789–805, 1992. 1

[8] C. Oden, A. Ercil, and B. Buke. Combining implicit polynomials and geometric features for hand recognition. *Pattern Recogn. Lett.*, 24(13):2145–2152, 2003. 1

[9] T. Sahin and M. Unel. Fitting globally stabilized algebraic surfaces to range data. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1083–1088, Washington, DC, USA, 2005. IEEE Computer Society. 2, 3

[10] T. Tasdizen, J. Tarel, and D. Cooper. Improving the stability of algebraic curves for applications. *IEEE Transactions on Image Processing*, 9(3):405–416, 2000. 2, 6, 7

[11] G. Taubin. Estimation of planar curves, surfaces, and non-planar 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. 1

[12] C. Unsalan. A model based approach for pose estimation and rotation invariant object matching. *Pattern Recogn. Lett.*, 28(1):49–57, 2007. 1

[13] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, and K. Ikeuchi. A fast registration method using ip and its application to ultrasound image registration. In *IPSA Transactions on Computer Vision and Applications*, pages 209–219, September 2009. 1

[14] B. Zheng, J. Takamatsu, and K. Ikeuchi. 3d model segmentation and representation with implicit polynomials. *IEICE Transactions*, 91-D(4):1149–1158, 2008. 1