IMPLICIT B-SPLINE FITTING USING THE 3L ALGORITHM

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 proposes a novel extension of the 3L algorithm to the B-Splines solution space. The 3L algorithm is a fast algebraic method for fitting a set of points through implicit curves or surfaces. It was originally proposed for Implicit Polynomials, which although simple and attractive are not flexible representations. In this paper Implicit B-Splines (IBSs) are used to define the solution space of the 3L algorithm. IBSs offer flexible representations, which can be locally controlled. These properties are exploited for regularizing the solution space. The experimental results illustrate that the proposed framework outperforms previous formulation.

***Index Terms*—** Curve and surface fitting; Algebraic methods; Implicit B-Splines; Regularization.

1. INTRODUCTION

Surface reconstruction from cloud of 3D points is one of the major problems in 3D computer vision. It has been active for decades [1, 2, 3]; implicit representations are attractive and widely used in the literature. These models do not need any parametrization, and they are able to describe objects with complex topologies. Implicit Polynomial (IP) provides one of the simplest solution space to describe curves and surfaces; it is simply defined by a coefficient vector. The optimal set of coefficients can be found through either algebraic or geometric methods. The first category includes techniques based on least squares optimization trying to satisfy a set of constraints [4, 5, 6]. These methods are fast and easy to implement, but they miss accuracy in fitting, due to the lack of real distance meaning during the optimization process. The second category tackles the accuracy problem by considering the orthogonal distance from point to surface [1, 7]. These methods find the optimal coefficients by non-linear models like Quasi Newton or Levenberg-Marquadt algorithms.

The 3L algorithm belongs to algebraic methods for Implicit Polynomial fitting [4]. It works as follows: firstly, two additional offsets from inside and outside of the boundary must be constructed. Then, the problem is modelled so that the optimal IP approaches zero in the original set and obtains sign transition from inside to outside. Some additional constraints like spatial control points could be imposed on in order to increase the fitting precision.

Implicit Polynomials are linearly defined from the coefficient vector, and each coefficient has a global affection on the shape. Hence, changes in one parameter may lead to a change in the whole shape. This problem is tackled in [5] in order to regularize the optimal IP coefficients from the 3L algorithm. Another solution is to use a different representation model than IP [8, 9].

Implicit B-Spline (IBS) is an implicit representation that is still linearly described with respect to the control parameters. Moreover, each parameter has a local contribution to the shape. This property makes it useful for algebraic fitting methods, although it can be also used in the geometric framework [2, 10]. In addition, some regularization constraints can be easily imposed in order to control the global shape of the object.

The authors in [9] propose to use IBSs to fit the given data points with the associated normals. Their method is similar to the gradient-one algorithm [11], which tries to maintain the normal directions while fitting the given data points. Moreover, a global tension term is used in order to regularize the optimal IBS parameters. The implicit fitting result can be modified by a parametric fitting through a dual evolution approach [12].

In the current work the 3L algorithm is extended for implicit B-Spline curves and surfaces. In the next section the 3L algorithm is firstly introduced for IP space; then the proposed extension for IBSs is presented. Additionally, some techniques are proposed in order to regularize the control lattice. The experimental results in section 3 show the advantages of IBSs over IPs, both in flexibility and regularization. Finally, conclusions are presented in section 4.

2. PROPOSED APPROACH

This section presents the proposed extension of the 3L algorithm to tackle the IBS parameter estimation problem. Firstly, the 3L algorithm, for implicit polynomial case, is explained. Then, in order to compensate the lack of flexibility of this representation, IBS is introduced as another solution space. Next, the 3L algorithm is adapted for this new space. Finally, the regularization of IBS parameters is presented in order to control the whole shape variation. This stage includes the parameter refinement after the optimization, and imposes additional constraints during the optimization.

2.1. THE 3L FORMULATION

The fitting problems try to find a set of parameters describing the given set of points $\Gamma_0 = \{(x_k, y_k)\}_1^{N_d}$. These parameters could chosen as the coefficients of implicit polynomials defined as:

$$f(x,y) = \sum_{i+j<=n} c_{i,j} x^i y^j, \tag{1}$$

which is made by basis functions $\{1, x, y, x.y, ..., x.y^{n-1}, y^n\}$. Each basis function has a support over $\Re$, so each coefficient has

This work has been partially supported by the project TRA2010-21371-C03-01 and research programme Consolider-Ingenio 2010: MIPRCV (CSD2007-00018).

a contribution on the whole region. The coefficients $c_{i,j}$ must be found such that the values of f in the given data set get close to zero. Then, the zero set of f defined by $Z_f = \{(x, y) \in \Re^2 | f(x, y) = 0\}$ describes the data set in a compact way.

In order to compensate the lack of geometric meaning, and to solve the instability problem in the classical algebraic methods [1], the authors in [4] have proposed the 3L algorithm, which consists in generating two additional *level sets*: $\Gamma_{-\delta}$ and $\Gamma_{+\delta}$ from the original data set Γ_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. Having considered these constraints, the 3L algorithm results in a more stable solution.

In more details, the 3L fitting algorithm is formalized as a linear least squares *explicit* polynomial fitting problem. Considering the three level sets: $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$ the over-determined system $\mathbf{M}_{3L}\mathbf{a} = \mathbf{b}$ must be solved, for the block matrix $\mathbf{M}_{3L}$ and the column vector $\mathbf{b}$:

$$\mathbf{M}_{3L} = \begin{bmatrix} \mathbf{M}_{\Gamma_{-\delta}} \\ \mathbf{M}_{\Gamma_0} \\ \mathbf{M}_{\Gamma_{+\delta}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{0} \\ +\mathbf{c} \end{bmatrix}, \tag{2}$$

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:

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

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

2.2. IMPLICIT B-SPLINES

An implicit B-Spline is defined as a zero set of tensor product of B-Splines:

$$f(x, y) = \sum_{i=1}^{M}\sum_{j=1}^{N} p_{i,j}B_{i,j}(u, v), \tag{4}$$

where $p_{i,j}$ is the lattice of control coefficients, and $B_{i,j}(u, v) = b_i(u).b_j(v)$ is the tensor product of two spline basis functions with different parameters. The matrix $\{p_{i,j}\}_{M\times N}$ contains the parameters of IBS. On the contrary to IPs, each basis function in IBSs has a compact support (i.e, it obtains non-zero values in an interval and vanishes outside it). Figure 1 shows the cubic tensor product B-Spline defined on the unit square $[0, 1]^2$. As illustrated in the figure, each function is composed of different cubic patches defining the curve in C^2. The basis functions used for cubic IBS, which guarantee the C^2 continuity, are as follows:

$$b_0(u) = (1 - u)^3/6, \quad b_1(u) = (3u^3 - 6u^2 + 4)/6, \tag{5}$$
$$b_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6, \quad b_3(u) = u^3/6.$$

For every given point $(x, y) \in [0, X] \times [0, Y]$ its corresponding spline parameters $(u, v) \in [0, 1]^2$ are defined as:

$$u = \frac{x}{X}M - \lfloor\frac{x}{X}M\rfloor, \tag{6}$$
$$v = \frac{y}{Y}N - \lfloor\frac{y}{Y}N\rfloor.$$

Since an IBS is linearly described by control values (4) the least squares method could be used to find the best set of parameters. The only difference with IPs is the way to construct the monomial matrix M in the original data set, the inner and outer offsets. This matrix

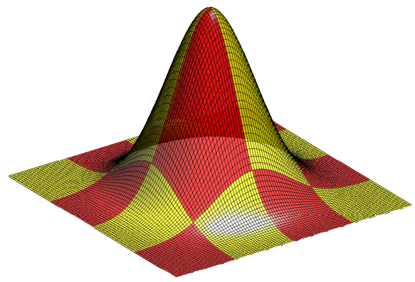

Fig. 1. A cubic IBS basis function defined on the unit square.

contains monomial vectors in each row showing the coefficients related to each parameter in the given point.

In order to handle the 3L algorithm, the control lattice $\{p_{i,j}\}$ must be converted to the vector form $\mathbf{p}$. The vector index corresponding to $p_{i,j}$ is $k = N.(i - 1) + j$. This relationship between two indexing must be kept everywhere: once in the monomial matrix computation, and once to convert the final result. The optimal vector can be found through least squares:

$$\mathbf{p} = \mathbf{M}_{3L}^{\dagger}\mathbf{b} = (\mathbf{M}_{3L}^T\mathbf{M}_{3L})^{-1}\mathbf{M}_{3L}^T\mathbf{b}, \tag{7}$$

where the monomial matrix M_{3L} contains the coefficients of each control parameter for each point. This matrix is constructed as follows: each rows of the matrix M_{3L} corresponds to a point (x, y), either from the original data set, or from one of the offsets. This point has a contribution on a local 4×4 net starting from indices: $i = \lfloor\frac{x}{X}.M\rfloor$ and $j = \lfloor\frac{y}{Y}.N\rfloor$. Traversing all the active parameters $\{p_{i+m,j+n}\}_{m,n\in\{0,1,2,3\}}$, the value $b_m(u).b_n(v)$ must be saved in the column $k = N.(i+k-1)+(j+l)$ and the current row of M_{3L}.

2.3. REGULARIZATION

The matrix in (7) might be singular, which leads to more than one solution as an optimal IBS. In fact, during the least squares optimization some of the control parameters do not have any contribution. So, they are not taken into account during the minimization, and it leads to a subspace of solutions.

Using a global tension term is a common method of parameter regularization [9]. This term is computed by measuring the curvature of f over the whole domain:

$$T(\mathbf{p}) = \int\int_D f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dx.dy. \tag{8}$$

This term is the same one used as a bending energy in Thin Plate Spline (TPS) for deformations. It must be emphasized that this term can be analytically computed for the given control parameter. This is due to the linear definition of IBSs, and the linearity of the integral operation. Since it is a quadratic term, its derivative will be still linear, and the final solution could be solved through a linear system of equations [9].

In the current work, we use Ridge Regression (RR) which is a widely used method for regularizing ill-posed problems [5]. In this technique the monomial matrix will be added to a diagonal matrix in order to obtain a nonsingular one:

$$\mathbf{p} = (\mathbf{M}_{3L}^T\mathbf{M}_{3L} + \lambda.diag)^{-1}\mathbf{M}_{3L}^T\mathbf{b}. \tag{9}$$

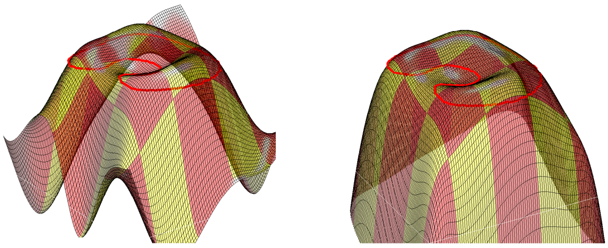

Fig. 2. The 3L algorithm for 2D fitting though an IBS $(left)$ before, and $(right)$ after editing the control parameters.

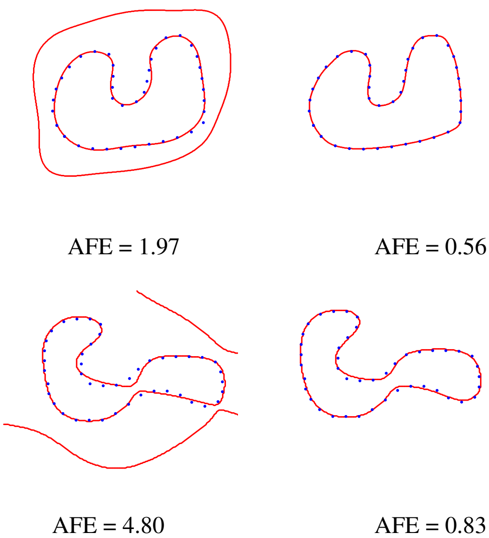

Fig. 3. Fitting 2D set of points with: $(left)$ 6th degree IPs; $(right)$ proposed IBSs.

The diagonal matrix, $diag$, could be either an identity matrix or the diagonal matrix of $\mathbf{M}_{3L}^T\mathbf{M}_{3L}$ or a combination of both of them.

In addition, a coarse to fine regularizing technique could be applied. For this purpose, we start from a low resolution control lattice. Having computed the solution, these control values could be imposed as new constraints on the higher resolution control lattice. Moreover, after having the final solution, it could be easily pruned through substituting the values of inactive control parameters by some constant values. These constant values must be a large positive or a small negative to show the inside and outside respectively.

3. EXPERIMENTAL RESULTS

The proposed approach has been validated for data sets in 2D and 3D. The 3L algorithm has been used in all examples, once for implicit polynomials (IPs), and once for implicit B-Splines (IBSs). The accuracy in both cases is quantitatively evaluated using the *fitting error* (FE) computed for every single point with [7]. These values are used to obtain a quantitative criterion for comparison, which is

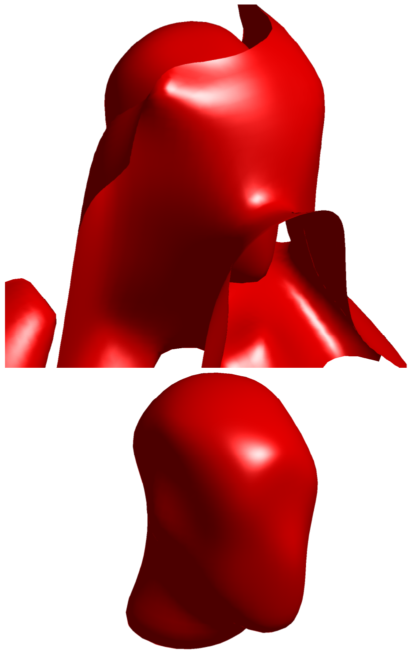

Fig. 4. The result of 3L algorithm for a set of 3D points through: (top) a 9th degree IP (AFE=4.20) and $(bottom)$ an IBS obtained with the proposed approach (AFE=3.08).

referred to as *Accumulated Fitting Error* (AFE): AFE= $\sum_{i=1}^{N_d} FE_i$. The advantage of IBSs as a solution space for 3L is illustrated in these examples. Figure 2$(left)$ shows the output of the 3L algorithm applied in IBS space. As illustrated in this figure, the control parameters around the domain border need to be regularized. For this purpose we only changed the control parameters away from the object. These parameters do not have any contribution for the points around the given data set. Figure 2$(right)$ depicts the improvement after editing the inactive control parameters. Figure 3 compares the result of 3L-IP and 3L-IBS for two sets of 2D points. As illustrated in this figure, IBS solution space can be easily controlled, and reach a better fitting result without outliers.

In the current work, the 3L-IBS is mainly proposed for the 3D point fitting, when the final implicit surface is self-occluded. Figure 4 (top) depicts the result of the 3L algorithm for an nineth degree IP. Since the 3L algorithm just imposes some constraints around the given data, there is no control away from the object. Moreover, since the coefficient vector of IP has a global affection on the shape, it is not easy to edit these parameters after the fitting stage. This problem is solved through using a $12 \times 12 \times 12$ IBS (Fig. 4 $(bottom)$). The inactive control parameters, which are away from the object, are set to a constant value in order to avoid any sign transition in the implicit function. Similarly, Fig. 5 compares the result of 3L-IP for a tenth degree IP and 3L-IBS for a $14 \times 14 \times 14$ IBS. The same order IBS is used in Fig. 6 $(bottom)$ while an eighth degree IP is used for the 3L-IP in Fig. 6 (top). It must be mentioned that using a higher degree results in a more unstable IP. The only way to edit the coefficient vector is to change the 3L model, and to add some regularization term inside the model like RR in [5], which decreases the fitting precision [6]. The advantage of IBS to IP is due to the facilities provided for parameter regularization during the fitting, and parameter editing even after the fitting.

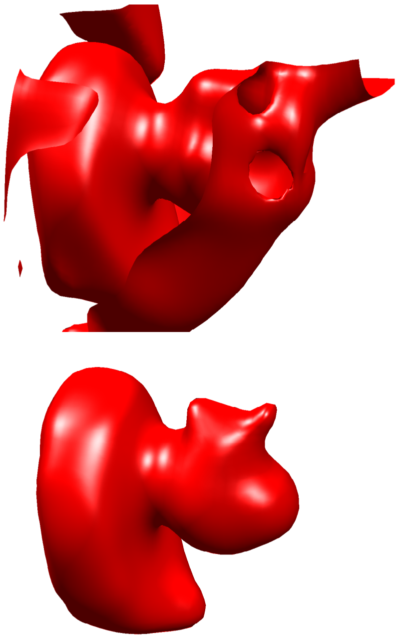

Fig. 5. The result of 3L algorithm for a set of 3D points through: (*top*) a 10th degree IP (AFE=2.83) and (*bottom*) an IBS obtained with the proposed approach (AFE=2.04)

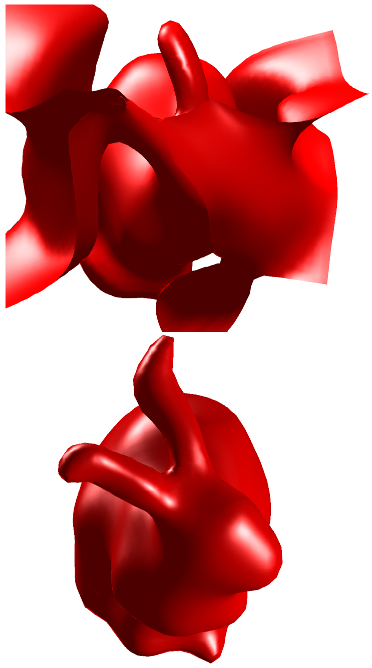

Fig. 6. The result of 3L algorithm for Bunny through: (*top*) a 8th degree IP (AFE=5.92) and (*bottom*) an IBS obtained with the proposed approach (AFE=4.57).

4. CONCLUSIONS

In this work, IBSs are used to define the solution space of the 3L algorithm instead of IPs, which were originally used. An IBS is a linear combination of the control parameters; moreover, each parameter has a local contribution on the shape, which makes it more desirable than IPs. The experimental results show the advantages of IBSs, and the facility for parameter editing. Future work will be focused on the application of this flexible representation for tackling the registration problem.

5. REFERENCES

[1] 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, 1991.

[2] M. Aigner and B. Jüttler, "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.

[3] 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.

[4] 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.

[5] 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*, Washington, DC, USA, 2005, pp. 1083–1088, IEEE Computer Society.

[6] Mohammad Rouhani and Angel 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.

[7] 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.

[8] B. Morse, T. Yoo, D. Chen, P. Rheingans, and K. Subramanian, "Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions," in *Shape Modeling International*, 2001, pp. 89–98.

[9] B. Jüttler and A. Felis, "Least-squares fitting of algebraic spline surfaces," *Adv. Comput. Math.*, vol. 17, no. 1-2, pp. 135–152, 2002.

[10] Z. Yang, J. Deng, and F. Chen, "Fitting unorganized point clouds with active implicit b-spline curves," *The Visual Computer*, vol. 21, no. 8-10, pp. 831–839, 2005.

[11] 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.

[12] R. Feichtinger, M. Fuchs, B. Jüttler, O. Scherzer, and Huaiping Yang, "Dual evolution of planar parametric spline curves and t-spline level sets," *Computer-Aided Design*, vol. 40, no. 1, pp. 13–24, 2008.