Adaptively Determining Degrees of Implicit Polynomial Curves and Surfaces

Bo Zheng, Jun Takamatsu, and Katsushi Ikeuchi

Institute of Industrial Science, The University of Tokyo, Komaba 4-6-1, Meguro-ku, Tokyo, 153-8505 Japan
zheng@cvl.iis.u-tokyo.ac.jp

Abstract. Fitting an implicit polynomial (IP) to a data set usually suffers from the difficulty of determining a moderate polynomial degree. An over-low degree leads to inaccuracy than one expects, whereas an over-high degree leads to global instability. We propose a method based on automatically determining the moderate degree in an incremental fitting process through using QR decomposition. This incremental process is computationally efficient, since by reusing the calculation result from the previous step, the burden of calculation is dramatically reduced at the next step. Simultaneously, the fitting instabilities can be easily checked out by judging the eigenvalues of an upper triangular matrix from QR decomposition, since its diagonal elements are equal to the eigenvalues. Based on this beneficial property and combining it with Tasdizen's ridge regression method, a new technique is also proposed for improving fitting stability.

1 Introduction

Recently representing 2D and 3D data sets with implicit polynomials (IPs) has been attractive for vision applications such as fast shape registration, pose estimation [1,2,3,4], recognition [5], smoothing and denoising, image compression [6], etc. In contrast to other function-based representations such as B-spline, Non-Uniform Rational B-Splines (NURBS), and radial basis function (RBF) [7], IPs are superior in the areas of fast fitting, few parameters, algebraic/geometric invariants, robustness against noise and occlusion, etc.

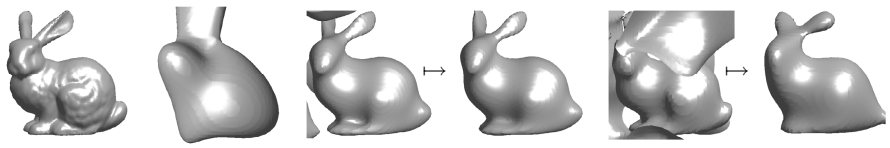
A 3D IP function of degree n is defined as:

$$f_n(\boldsymbol{x}) = \sum_{0\le i,j,k;i+j+k\le n} a_{ijk}x^i y^j z^k = (\underbrace{1 \;\; x \;\; \ldots \;\; z^n}_{\boldsymbol{m}(\boldsymbol{x})^T})(\underbrace{a_{000} \;\; a_{100} \;\; \ldots \;\; a_{00n}}_{\boldsymbol{a}})^T, \quad (1)$$

where $\boldsymbol{x} = (x \;\; y \;\; z)$ is a data point. $f_n(\boldsymbol{x})$'s zero set $\{\boldsymbol{x}|f_n(\boldsymbol{x}) = 0\}$ is used to represent the given data set. The estimation of IP coefficients belongs to the conventional fitting problem, and various methods have been proposed [1,2,3,4,8].

These fitting methods cannot adaptively control the IP degrees for different object shapes; it is well known that simple shapes correspond to low-degree IPs whereas complicated shapes correspond to the higher ones. Fig.1 shows that

Y. Yagi et al. (Eds.): ACCV 2007, Part II, LNCS 4844, pp. 289–300, 2007.



(a) (b) (c) (c') (d) (d')

Fig. 1. IP fitting results: (a) Original data set; (b) 4-degree IP surface; (c) 8-degree IP surface; (c') Stability-improved 8-degree IP surface; (d) 10-degree IP surface; (d') Stability-improved 10-degree IP surface

when fitting an object like Fig.1(a), an over-low degree leads to the inaccuracy (Fig.1(b)), whereas an over-high degree leads to an unstable result: too many unexpected zero sets appear (see Fig.1(d)). This paper provides a solution to automatically find the moderate degree IP (such as Fig.1(c)).

Another issue of IP fitting is that there may be collinearity in the covariant matrix derived from the least squares method, making them prone to instability [4], e.g., the fitting results shown in Fig.1(c) and (d). In order to address this issue we propose a method for automatically checking out this collinearity and improving it. And we also combine the Ridge Regression (RR) technique recently introduced by [4,9]. Fig.1(c') and (d') show the improved results of Fig.1(c) and (d) respectively, where the redundant zero sets are retired. Note although Fig.1 (d) is globally improved by our method to Fig.1 (d'), since there are too many redundant zero sets that need to be eliminated, the local accuracy is also affected very much. Therefore, we first aim at adaptively finding a moderate degree, and then applying our stability-improving method to obtain a moderate result (accurate both locally and globally).

This paper is organized as follows: Section 2 gives a review of IP fitting methods; Section 3 and 4 provide an incremental method for fitting IP with moderate degrees; Section 5 discusses on how to improve the global stability based on our algorithm; Section 6 presents experimental results followed by discussion and conclusions in section 7 and 8.

2 Implicit Polynomial Fitting

In order to estimate the coefficient vector $\boldsymbol{a}$ in (1), a simple method is to solve a linear system

$$M\boldsymbol{a} = \boldsymbol{b}, \tag{2}$$

where M is the matrix of monomials, and the ith row of M is $\boldsymbol{m}(\boldsymbol{x_i})$ (see (1)) and $\boldsymbol{b}$ is a zero vector. But generally M is not a square matrix, and the linear system is usually solved by the least squares method.

Solutions to this least squares problem are classified into nonlinear methods [1,2,10] and linear methods [3,4,8,9]. Because the linear methods are

simpler and much faster than the nonlinear ones, they are preferable and can be formulated as:

$$M^T M\boldsymbol{a} = M^T\boldsymbol{b}. \tag{3}$$

Note this formula is just transformed from the least squares result, $\boldsymbol{a} = M^\dagger\boldsymbol{b}$, where $M^\dagger = (M^T M)^{-1}M^T$ called a pseudo-inverse matrix.

Direct solution of the linear system is numerically unstable, since $M^T M$ is nearly singular and $\boldsymbol{b} = \boldsymbol{0}$. Thus $\boldsymbol{a}$ is determined from the kernel basis. Fortunately, the methods for improving the classical linear fitting (avoiding the ill-condition matrix $M^T M$ in (3)) have already been proposed by adding some constraints, such as the 3L method [3], the Gradient-One method [4] and the Min-Max method [8]. The singularity of M is improved and $\boldsymbol{b}$ is also derived as a nonzero vector.

In the prior methods, the symmetric linear system (3) was solved by the classical solvers such as the Cholesky decomposition method, the conjugate gradient method, singular value decomposition (SVD), and their variations. But none of them allow changing the degree during the fitting procedure. This is the main reason why these prior methods require a fixed degree.

3 Incremental Fitting

This section shows computational efficiency of the proposed incremental fitting method. Although an IP degree is gradually increased until obtaining a moderate fitting result, the computational cost is saved because each step can completely reuse the calculation results of the previous step.

In this section, first we describe the method for fitting an IP with the QR decomposition method. Next, we show the incrementability of Gram-Schmidt QR decomposition. After that, we clarify the amount of calculation in order to increase the IP degree.

3.1 Fitting Using QR Decomposition

Without solving the linear system (3) directly, our method first carries out QR decomposition on matrix M as $M = Q_{N\times m}R_{m\times m}$, where Q satisfies the $Q^T Q = I$ (I is the identity matrix), and R is an invertible upper triangular matrix.

Then substituting $M = QR$ into (2), we obtain:

$$QR\boldsymbol{a} = \boldsymbol{b} \rightarrow Q^T QR\boldsymbol{a} = Q^T\boldsymbol{b} \rightarrow R\boldsymbol{a} = Q^T\boldsymbol{b} \rightarrow R\boldsymbol{a} = \widetilde{\boldsymbol{b}}. \tag{4}$$

Since R is an upper triangular matrix, $\boldsymbol{a}$ can be quickly solved (in $\mathcal{O}(m^2)$).

3.2 Gram-Schmidt QR Decomposition

Let us assume that matrix M consisting of columns $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_m\}$ is known. We show the method of Gram-Schmidt orthogonalization, that is, how to orthogonalize the columns of M into the orthonormal vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m\}$ which

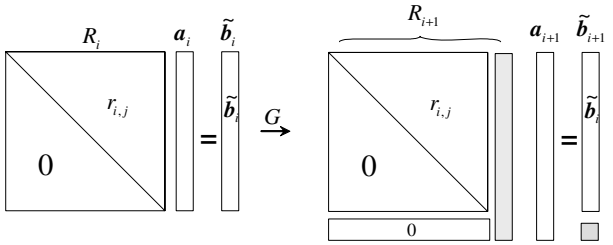


Fig. 2. The triangular linear system grows from the ith step to the $(i+1)$th, and only the calculation shown in light-gray is required

are columns of matrix Q, and simultaneously calculate the corresponding upper triangular matrix R consisting of elements $r_{i,j}$. The algorithm is as follows: Initially let $\boldsymbol{q}_1 = \boldsymbol{c}_1 / \parallel \boldsymbol{c}_1 \parallel$ and $r_{1,1} = \parallel \boldsymbol{c}_1 \parallel$. If $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_i\}$ have been computed, then the (i+1)th step is:

$$
\begin{aligned}
& r_{j,i+1} = \boldsymbol{q}_j^T \boldsymbol{c}_{i+1}, \quad \text{for } j \leq i, \\
& \boldsymbol{q}_{i+1} = \boldsymbol{c}_{i+1} - \sum_{j=1}^{i} r_{j,i+1} \boldsymbol{q}_j, \\
& r_{i+1,i+1} = \parallel \boldsymbol{q}_{i+1} \parallel, \\
& \boldsymbol{q}_{i+1} = \boldsymbol{q}_{i+1} / \parallel \boldsymbol{q}_{i+1} \parallel .
\end{aligned} \tag{5}
$$

From this algorithm, we can see that Gram-Schmidt orthogonalization can be carried out in an incremental manner, which orthogonalizes the columns of M one by one.

3.3 Additional Calculation for Increasing an IP Degree

The incrementability in the QR decomposition with Gram-Schmidt orthogonalization makes our incremental method computationally efficient. Fig.2 illustrates this efficiency by the calculation from the ith step to the $(i+1)$th step in our incremental process. It is only necessary to calculate the parts shown in light-gray. For constructing these two upper triangular linear systems from the ith step to the $(i+1)$th step, we only need two types of calculation: 1) for growing the upper triangular matrix from R_i to R_{i+1}, calculate the rightmost column and add it into the R_i to construct R_{i+1}, and 2) for growing the right-hand vector from $\widetilde{\boldsymbol{b}}_i$ to $\widetilde{\boldsymbol{b}}_{i+1}$, calculate the bottom element and add it into the $\widetilde{\boldsymbol{b}}_i$ to construct $\widetilde{\boldsymbol{b}}_{i+1}$. For the first calculation, it can be simply obtained from Gram-Schmidt orthogonalization in (5). For the second calculation, assuming $\widetilde{b}_{i+1}$ is the bottom element of vector $\widetilde{\boldsymbol{b}}_{i+1}$, the calculation of $\widetilde{b}_{i+1}$ can follow the $(i+1)$th step of Gram-Schmidt orthogonalization in (5), and can be calculated as $\widetilde{b}_{i+1} = \boldsymbol{q}_{i+1}^T \boldsymbol{b}$.

In order to clarify the computational efficiency, let us assume a comparison between our method and an incremental method that iteratively calls a linear

method such as the 3L method at each step. It is obvious that, for solving coefficient $\boldsymbol{a}$ at the ith step, our method needs i inner-product operations for constructing the upper triangular linear system, and $\mathcal{O}(i^2)$ for solving this linear system; whereas the latter method needs $i \times i$ inner-product operations for constructing linear system (3), and $\mathcal{O}(i^3)$ for solving (3).

Let us define a function G to denote the above calculation. Then if we repeat calling function G, we can obtain the incremental (dimension-growing) upper triangular linear systems, and the corresponding coefficient vectors with different degrees can be solved from them.

4 Finding the Moderate Degree

Now to construct an algorithm for finding the moderate degree, we are facing two problems:

1) What is the order for choosing $\boldsymbol{c}_\alpha$?
2) When should the incremental procedure be stopped?

Note: as a matter of convenience, hereafter we use notation α to denote the column index of M instead of i.

4.1 Incremental Order of Monomials

Feeding the columns $\boldsymbol{c}_\alpha$ from M into the function G in a different order may lead to the different result. Therefore it is important to decide a suitable order. A reasonable way is to choose $\boldsymbol{c}_\alpha$ in the degree-growing order described by Tab.1. The reason is as follows: when we fit a two-degree IP to the data on a unit circle, a unique solution $-1 + x^2 + y^2 = 0$ can be obtained, while if we choose a three-degree IP to fit, solutions such as $x(-1 + x^2 + y^2) = 0$, are obtained. There exist some redundant zero set groups, such as $x = 0$.

4.2 Stopping Criterion

For the second problem, we have to define a stopping criterion based on our defined similarity measure between IP and data set. Once this stopping criterion is satisfied, we consider the desired accuracy is reached and the procedure is terminated. First let us introduce a set of similarity functions measuring the similarity between IP and data set, as follows:

$$D_{dist} = \frac{1}{N}\sum_{i=1}^{N} e_i, \quad D_{smooth} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{N}_i \cdot \boldsymbol{n}_i), \tag{6}$$

where N is the number of points, $e_i = \frac{|f(\boldsymbol{x}_i)|}{\|\bigtriangledown f(\boldsymbol{x}_i)\|}$, $\boldsymbol{N}_i$ is the normal vector at a point obtained from the relations of the neighbor normals (here we refer to Sahin's method [9]), and $\boldsymbol{n}_i = \frac{\bigtriangledown f(\boldsymbol{x}_i)}{\|\bigtriangledown f(\boldsymbol{x}_i)\|}$ is the normalized gradient vector of f at $\boldsymbol{x}_i$. e_i has proved useful for approximating the Euclidean distance from $\boldsymbol{x}_i$ to the IP zero set [2].

Table 1. Index List: i, j and k are the powers of x, y and z respectively. α is the index of column of M. And the relations between α and (i, j, k) can be formulated as: $\alpha = j + (i+j+1)(i+j)/2 + 1$ (for 2D) and $\alpha = k + (j+k+1)(j+k)/2 + (i+j+k+2)(i+j+k+1)(i+j+k)/6 + 1$ (for 3D).

α	$[i\ j]$	Form
1	[0 0]	L0 $(i+j=0)$
2	[1 0]	
3	[0 1]	L1 $(i+j=1)$
4	[2 0]	
5	[1 1]	
6	[0 2]	L2 $(i+j=2)$
7	[3 0]	
8	[2 1]	
9	[1 2]	
10	[0 3]	L3 $(i+j=3)$
	⋮	
m	[0 n]	Ln $(i+j=n)$

(a) Index list for 2D

α	$[i\ j\ k]$	Form
1	[0 0 0]	L0 $(i+j+k=0)$
2	[1 0 0]	
3	[0 1 0]	
4	[0 0 1]	L1 $(i+j+k=1)$
5	[2 0 0]	
6	[1 1 0]	
7	[0 2 0]	
8	[1 0 1]	
9	[0 1 1]	
10	[0 0 2]	L2 $(i+j+k=2)$
	⋮	
m	[0 n]	Ln $(i+j+k=n)$

(b) Index list for 3D

D_{dist} and D_{smooth} can be considered as two measurements on distance and smoothness between data set and IP zero set. And we define our stopping criterion as:

$$(D_{dist} < T_1) \ \wedge \ (D_{smooth} > T_2). \tag{7}$$

4.3 Algorithm for Finding the Moderate IPs

Having the above conditions, our algorithm is simply described as follows:

1) Calling the function G to construct the upper triangular linear system;
2) Solving this linear system to obtain coefficient vector $\boldsymbol{a}$;
3) Measuring the similarity for the obtained IP;
4) Stopping the algorithm if the stopping criterion (7) is satisfied; otherwise going to 1) for growing up the dimension.

5 Improving Global Stability

Linear fitting methods in general suffer from not achieving global stability, which is well discussed in [4,9]. Since our fitting method belongs to these linear methods, we face the same problem. We propose a method for solving this by controlling the condition number of matrix M.

5.1 Stability and Condition Number of M

An important reason for global instability is the collinearity of M, which causes the matrix $M^T M$ to be nearly singular with some eigenvalues negligible

compared to the others [4]. Such eigenvalues are degenerated to contribute very little to the overall shape of the fit.

Fortunately, since $M = QR$, we can obtain $M^TM = R^TR$, and thus the condition number of M^TM can be improved by controlling the eigenvalues of R. Note here we just consider the condition number as $\lambda_{max}/\lambda_{min}$, where λ_{max} and λ_{min} are the maximum and minimum eigenvalues respectively. And from the good properties that upper triangular matrix R's eigenvalues lay on its main diagonal, we can easily evaluate the singularity of R by only observing the diagonal values. To improve the condition number of M^TM, this paper gives a solution from two aspects: eliminating collinear columns of M and using the Ridge Regression method.

5.2 Eliminating Collinear Columns of M

The first simple idea is to check the eigenvalue $r_{i,i}$ to see whether it is too small (nearly zero), at each step. If $r_{i,i}$ is too small, that means the current column $\boldsymbol{c}_i$ of M is nearly collinear to the space of $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_{i-1}\}$. Thus to find a viable value for R, it should be abandoned, and the subsequent columns should be tried.

5.3 Ridge Regression Method

Ridge regression (RR) regularization is an effective method that improves the condition number of M^TM by adding a small moderate value to each diagonal element, e.g., adding a term κD to M^TM [4,9]. Accordingly equation (3) can be modified as $(M^TM + \kappa D)\boldsymbol{a} = M^T\boldsymbol{b}$ and equation (4) can be modified as

$$(R + \kappa R^{-T} D)\boldsymbol{a} = \widetilde{\boldsymbol{b}} \tag{8}$$

where κ is a small positive value called the RR parameter and D is a diagonal matrix. D will be chosen to maintain Euclidean invariance, and the simplest choice is to let D be an identity matrix. A cleverer choice has been proposed by T.Tasdizen et al. [4] for 2D and T.Sahin et al. [9] for 3D. In fact, their strategies are to add the constraints that keep the leading forms of this polynomial always strictly positive, which proves that the zero set of polynomials with an even degree are always bounded (see the proof in [4]). We give details of this derivation in the appendix.

6 Experimental Results

The setting for our experiments involve some pre-conditions. 1) As a matter of convenience, we employ the constraints of the 3L method [3]; 2) All the data sets are regularized by centering the data-set center of mass at the origin of the coordinate system and scaling it by dividing each point by the average length from points to origin; 3) We choose T_1 in (7) with about 20 percent of the layer distance c of the 3L method as done in [3].

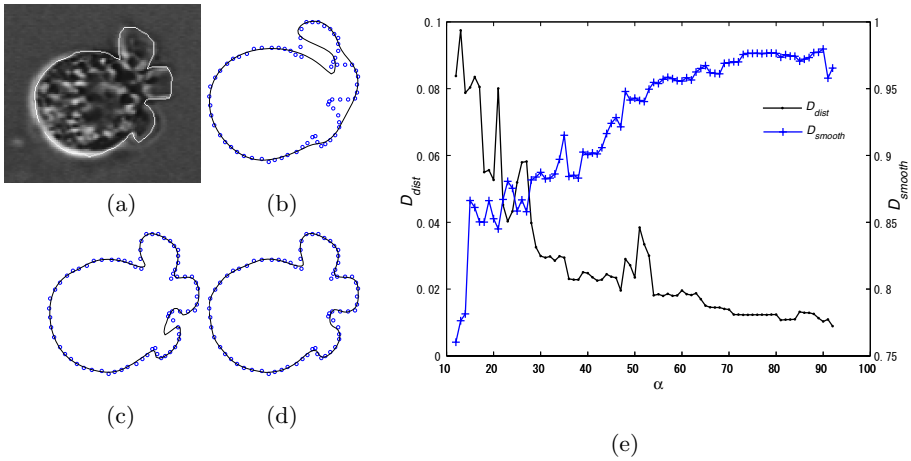


Fig. 3. IP fitting results: (a) Original image. (b) $\alpha = 28$ (six-degree). (c) $\alpha = 54$ (nine-degree). (d) $\alpha = 92$ (thirteen-degree). (e) Convergence of D_{dist} and D_{smooth}. Note "o" symbols represent the boundary points extracted from the image and real lines represent the IP zero set in (b)-(d).

6.1 A Numerical Example

In this experiment, we fit an IP to the boundary of a cell shown in Fig.3 (a). The stopping criterion is set as $T_1 = 0.01, T2 = 0.95$, and the layer distance of the 3L method is $c = 0.05$. The moderate IP is found out automatically (see Fig.3(d)). To give a comparison, we also show some fits before the desired accuracy is reached (see Fig.3(b) and (c)). And these results are improved by our method mentioned in section 5.

We also track the convergence of D_{dist} and D_{smooth} shown in Fig.3(e). Although there are some small fluctuations on the graph, D_{dist} and D_{smooth} are convergent to 0 and 1 respectively, which also proves the stopping criterion in (7) can effectively measure the similarity between the IP and the data set.

6.2 2D and 3D Examples

Some 2D and 3D experiments are shown in Fig.4 where the fitting results are obtained with the same parameters as those in the first example. As a conclusion here, objects with different shapes may obtain fitting results of different degrees, since these results always respect the complexity of shapes.

6.3 Degree-Fixed Fitting Compared with Adaptive Fitting

Fig.5 shows some comparisons between degree-fixed fitting methods and our adaptive fitting method. Compared to degree-fixed methods such as [3,4,8], the results of our method show that there is neither over-fitting nor the insufficient

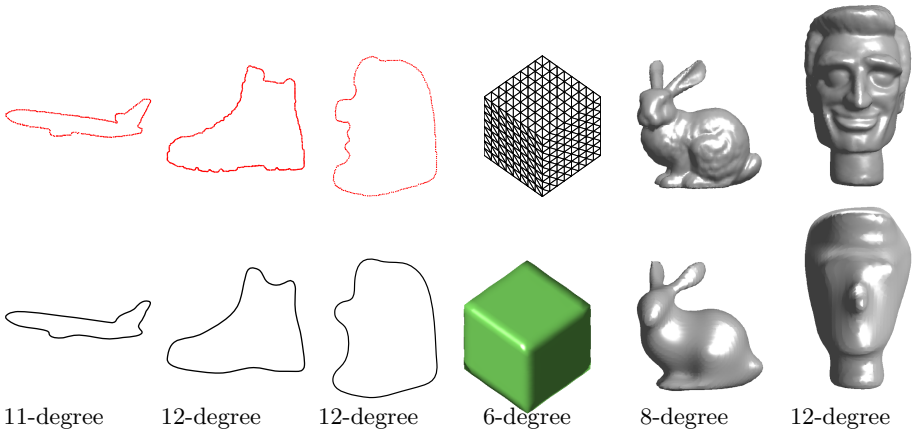


Fig. 4. IP fitting results. First row: Original objects; Second row: IP fits.

fitting, and we also attain global stability. It shows that our method is more meaningful than the degree-fixed methods, since it fulfills the requirement that the degrees should be subject to the complexities of object shapes.

7 Discussion

7.1 QR Decomposition Methods

Other famous algorithms of QR decomposition are Householder and Givens [11]. In the field of numerical computation, Householder and Givens have proved more stable than conventional Gram-Schmidt method. But in this paper, since our discussion is based on the good condition of a regularized data set, we ignore the small effect of rounding errors, which causes instability. Here we just would like to take advantage of the properties of QR decomposition that orthogonalize vectors one by one, to demonstrate the possibility of constructing the moderate-degree fitting algorithm described above.

7.2 IP vs Other Functions

In contrast to other function based representations such as B-spline, NURBS, and radial basis function, IP representation cannot give a relatively accurate model. But this representation is more attractive for applications that require fast registration and fast recognition (see the works [4,5,12,13,14]), because of its algebraic/geometric invariants [15]. Also, Sahin also showed some experiments for missing data in [9]. Further accurate representation of a complex object may require segmenting the object shapes and representing each segmented patch with an IP. We will consider this possibility in our future works.

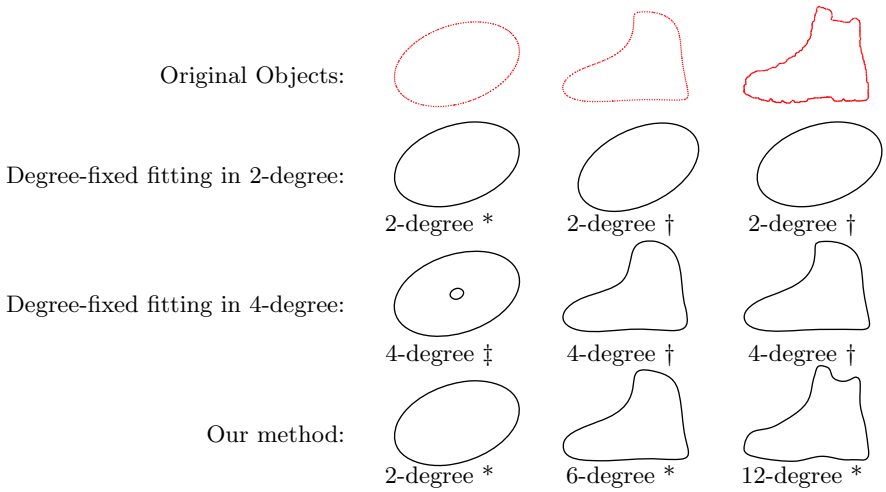


Fig. 5. Comparison between degree-fixed fitting and adaptive fitting. First row: Original objects. Second and third row: IP fits resulting from degree-fixed fitting with 2-degree and 4-degree fitting respectively. Fourth row: Adaptive fitting. Mark *: moderate fitting. †: insufficient fitting (losing accuracy). ‡: over-fitting.

7.3 Setting the Parameters T_1 and T_2

Since our stopping criterion can be approximately seen as a kind of Euclidean metric, it is intuitive to control moderate fitting accuracy by setting the appropriate values to T_1 and T_2. Basically, these parameters should be decided based on object scale or statistics about data noise. Further discussion is beyond the scope of this paper. Fortunately it is intuitive to decide if the parameters are appropriate for your demand, since the 2D/3D Euclidean distance can be easily observed. In this paper, we practically let T_1 and T_2 be close to zero and one respectively for a smooth model and more tolerant values for a coarse one.

8 Conclusions

This paper provided an incremental method for fitting shape-representing IPs. By our stopping criterion, an IP of moderate degree can be adaptively found in one fitting process, and global fitting stability is successfully improved. Our results support the argument that IP degrees being adaptively determined by shapes is better than being fixed, because this not only saves much time for users, but also it is suited to the future applications involving automatic recognition systems.

Acknowledgements

Our work was supported by the Ministry of Education, Culture, Sports, Science and Technology under the Leading Project: Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets.

References

1. Keren, D., Cooper, D.: Describing Complicated Objects by Implicit Polynomials. IEEE Trans. on Patt. Anal. Mach. Intell (PAMI) 16(1), 38–53 (1994)
2. 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 Patt. Anal. Mach. Intell (PAMI) 13(11), 1115–1138 (1991)
3. Blane, M., Lei, Z.B., Cooper, D.B.: The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data. IEEE Trans. on Patt. Anal. Mach. Intell (PAMI) 22(3), 298–313 (2000)
4. Tasdizen, T., Tarel, J.P., Cooper, D.B.: Improving the Stability of Algebraic Curves for Applications. IEEE Trans. on Imag. Proc. 9(3), 405–416 (2000)
5. Tarel, J.-P., Cooper, D.B.: The Complex Representation of Algebraic Curves and Its Simple Exploitation for Pose Estimation and Invariant Recognition. PAMI 22(7), 663–674 (2000)
6. Helzer, A., Bar-Zohar, M., Malah, D.: Using Implicit Polynomials for Image Compression. In: Proc. 21st IEEE Convention of the Electrical and Electronic Eng., pp. 384–388. IEEE Computer Society Press, Los Alamitos (2000)
7. Turk, G., OfBrieni, J.F.: Variational Implicit Surfaces. Technical Report GIT-GVU-99-15, Graphics, Visualization, and Useability Center (1999)
8. Helzer, A., Barzohar, M., Malah, D.: Stable Fitting of 2D Curves and 3D Surfaces by Implicit Polynomials. IEEE Trans. on Patt. Anal. Mach. Intell (PAMI) 26(10), 1283–1294 (2004)
9. Sahin, T., Unel, M.: Fitting Globally Stabilized Algebraic Surfaces to Range Data. In: Proc. 10th IEEE Int. Conf. on Compter Vision (ICCV), vol. 2, pp. 1083–1088 (2005)
10. Knanatani, K.: Renormalization for Computer Vision. The Institute of Elec., Info. and Comm. eng (IEICE) Transaction 35(2), 201–209 (1994)
11. Horn, R.A., Johnson, C.R.: Matrix Analysis: Section 2.8. Cambridge University Press, Cambridge (1985)
12. Tarel, J.P., Civi, H., Cooper, D.B.: Form 3d objects without point matching using algebraic surface models. In: Proceedings of IEEE Workshop Model Based 3D Image Analysis, pp. 13–21. IEEE Computer Society Press, Los Alamitos (1998)
13. Khan, N.: Silhouette-Based 2D-3D Pose Estimation Using Implicit Algebraic Surfaces. Master Thesis in Computer Science, Saarland University (2007)
14. Unsalan, C.: A Model Based Approach for Pose Estimation and Rotation Invariant Object Matching. Pattern Recogn. Lett. 28(1), 49–57 (2007)
15. Taubin, G., Cooper, D.: Symbolic and Numerical Computation for Artificial Intelligence. In: Computational Mathematics and Applications. ch. 6, Academic Press, London (1992)

Appendix

A. Choosing Diagonal Matrix *D* for RR Method

A choice of diagonal matrix D for RR method was derived by T.Tasdizen et al. [4] and T.Sahin et al. [9] as: $d_{\alpha\alpha} = \gamma\hat{t}$, where $d_{\alpha\alpha}$ is the αth diagonal element of D, and calculating $d_{\alpha\alpha}$ is respected to i, j, k. The relationship between index α and i, j, k are shown in Tab.1.

γ is a free parameter for the $(i + j)$th form (2D) or $(i + j + k)$th form (3D) decided from the data set as follows:

$$\gamma_{i+j} = \sum_{t=1}^{N}(x_t^2 + y_t^2)^{i+j} \quad (2D), \quad \gamma_{i+j+k} = \sum_{t=1}^{N}(x_t^2 + y_t^2 + z_t^2)^{i+j+k} \quad (3D), \tag{9}$$

where $(x_t\ y_t)$ and $(x_t\ y_t\ z_t)$ are the data point. $\hat{t}$ is a variable respected to i, j, k and for maintaining Euclidean invariance it can be derived as:

$$\hat{t} = \frac{i!j!}{(i+j)!} \quad (2D) \text{ and } \quad \hat{t} = \frac{i!j!k!}{(i+j+k)!} \quad (3D). \tag{10}$$