An Adaptive and Stable Method for Fitting Implicit Polynomial Curves and Surfaces

Bo Zheng, Jun Takamatsu, and Katsushi Ikeuchi, *Fellow, IEEE*

***Abstract*— Representing 2D and 3D data sets with implicit polynomials (IPs) has been attractive because of its applicability to various computer vision issues. Therefore, many IP fitting methods have already been proposed. However, the existing fitting methods can be and need to be improved with respect to computational cost for deciding on the appropriate degree of the IP representation and to fitting accuracy while still maintaining the stability of the fit.**

We propose a stable method for accurate fitting that automatically determines the moderate degree required. Our method increases the degree of IP until a satisfactory fitting result is obtained. The incrementability of QR decomposition with Gram-Schmidt orthogonalization gives our method computational efficiency. Furthermore, since the decomposition detects the instability element precisely, our method can selectively apply ridge regression-based constraints to that element only. As a result, our method achieves computational stability while maintaining fitting accuracy. Experimental results demonstrate the effectiveness of our method compared with prior methods.

***Index Terms*— Fitting algebraic curves and surfaces, Implicit polynomial (IP), Implicit shape representation**

I. Introduction

REPRESENTING 2D and 3D data sets with implicit polynomials (IPs) is currently attractive for vision applications such as fast shape registration [3], [12], [29], [25], [26], [1], [28], [21], recognition [10], [25], [24], [19], [18], [33], [36], smoothing and denoising [28], [22], 3D reconstruction [9], image compression [6], and image boundary estimation [27]. In contrast to other function-based representations such as B-spline, Non-Uniform Rational B-Splines (NURBS) [20], Rational Gaussian [5], and radial basis function (RBF) [32], IPs are superior in such areas as fast fitting, few parameters, algebraic/geometric invariants, and robustness against noise and occlusion.

To achieve the representation of 2D/3D shapes using an IP, the optimization of using classical least squares methods that minimize the distance between the data set and the zero level set of polynomial is advocated in recent literature [23], [29], [31], [12], [2], [1], [15], [16], [17], [28], [7], [14], [37], [11]. Another fitting method in [35] has the potential of offering an accurate model by converting Fourier series to IP form. However, the prior methods greatly suffer from two major issues as follows.

1) **Degree-fixed fitting procedure**

Before handling a fitting procedure, the degree of IP first must be determined according to the complexity of the object, and the degree will be fixed during the procedure. This is no problem when fitting a simple object; for example, it is easy to determine a 2D IP of degree 2 to fit a 2D shape that looks like an ellipsoid.

B. Zheng and K. Ikeuchi are with Institute of Industrial Science, the University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, JAPAN. E-mail: {zheng, ki}@cvl.iis.u-tokyo.ac.jp

J. Takamatsu is with NAIST, 8916-5 Takayama, NARA 630-0192 JAPAN.

Manuscript received March 1, 2008; revised Dec. 8, 2008.

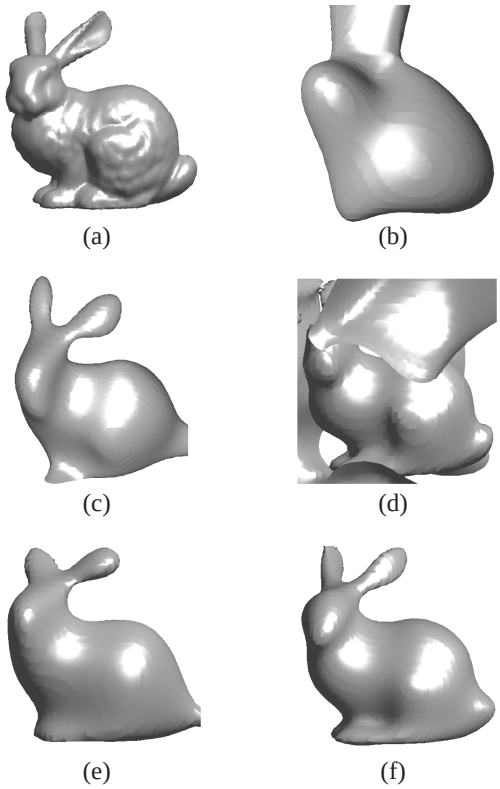

Fig. 1. IP fitting results of Original data (a): (b) 4-degree IP, (c) 6-degree IP, (d) 10-degree IP, (e) 10-degree IP stabilized by conventional RR regularization, (f) The result of our method.

However, the determination confuses a user who needs to fit a complex object such as the bunny in Fig. 1(a); determining an over-low degree leads to undesired inaccuracy (Fig. 1(b)), whereas determining an over-high degree leads to the over-fitting problem (Fig. 1(d)). Therefore, the user is forced to waste much time in finding a moderate degree by trying different degrees several times and selecting the best one from the results (Fig. 1(b)–(d)).

2) **Global fitting instability**

Despite the over-fitting problem, there is no doubt that the higher degree IP is very convenient for a complex object. Fortunately, several research groups have already proposed the method, *ridge regression (RR) regularization*, to overcome over-fitting problems, that is, to remove the extra zero sets by improving the numerical stability of the fitting procedure. Fig. 1(e) shows an example where the RR regularization was used to improve the global stability of the 10-degree IP shown in Fig. 1(d). The method is obviously useful for over-fitting, but, on the other hand, the method produces deterioration in local accuracy.

In this paper, we will solve these two issues with a novel

method based on QR-decomposition with Gram-Schmidt orthogonalization and a novel RR regularization. The method is capable of: 1) adaptively determining the moderate degree for fitting that depends on the complexity of objects, in an incremental manner without greatly increasing the computational cost; 2) retaining global stability while also avoiding excessive loss of local accuracy. As an example, shown in Fig. 1(f), our method adaptively determines an IP of moderate degree for fitting the *bunny* object, and the result shows better global and local accuracy than the prior method (Fig. 1(e)).

The first advantage of this method is its computational efficiency because the incrementability of Gram-Schmidt QR decomposition dramatically reduces the burden of the incremental fitting by reusing the calculation results of the previous step. The second advantage is that the method can successfully avoid global instability, and furthermore maintain local accuracy, because constraints derived from the RR regularization are selectively utilized in our incremental scheme. Finally, a set of stopping criteria can successfully stop the incremental scheme at the step where the moderate degree is achieved.

This paper is organized as follows: after reviewing some prior work on solving IP fitting problem in Section II, we present key components of our algorithm in Sections III, IV, and V. Section VI reports experimental results followed by discussion and conclusion in Sections VII and VIII.

II. Background

A. Mathematical Formulation

IP is the implicit function defined in a multivariate polynomial form. For example, the 3D IP of degree n is denoted by:

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

where $\mathbf{x} = (x\ y\ z)$ is one data point in the data set. An IP can be represented as an inner product between the monomial vector and the coefficient vector as $\mathbf{m}(\mathbf{x})^T\mathbf{a}$. The order for monomial indices $\{i, j, k\}$ is a degree-increasing order named *inverse lexicographical order* [30]. The *homogeneous binary polynomial* of degree r in x, y, and z, $\sum_{i+j+k=r} a_{ijk}x^i y^j z^k$, is called the r-th degree form of the IP.

B. Fitting Methods

The objective of IP fitting is to find a polynomial $f(\mathbf{x})$, of which the zero set $\{\mathbf{x}|f_n(\mathbf{x}) = 0\}$ can "best" represent the given data set. This fitting problem can be formulated in least squares optimization manner as:

$$M^T M\mathbf{a} = M^T\mathbf{b}, \quad (2)$$

where M is the matrix of monomials whose i-th row is $\mathbf{m}(\mathbf{x}_i)$ (see Eq. (1)); $\mathbf{a}$ is the unknown coefficient vector; and $\mathbf{b}$ is a zero vector. Note, Eq. (2) is just transformed from the least squares result, $\mathbf{a} = M^\dagger\mathbf{b}$, where $M^\dagger = (M^T M)^{-1}M^T$ is the so-called pseudo-inverse matrix.

Note, many efforts such as [1], [28], [7] have been made for solving the linear system of equations (2) by first overcoming the singularity of $M^T M$ and $\mathbf{b} = \mathbf{0}$. In this paper, we suppose that Eq. (2) has been modified by improving singularity of $M^T M$ and making $\mathbf{b} \neq \mathbf{0}$ by those techniques.

C. Ridge Regression (RR) Regularization for Stable Fitting

Although the linear methods improve the numerical stability to some extent, they still suffer from the difficulty of achieving global stability. Especially while modeling a complex shape with a high-degree IP, many extra zero sets are generated. One important reason for global instability is the collinearity of column vectors of matrix M, causing the matrix $M^T M$ to be nearly singular (see [28]).

Addressing this issue, Tasdizen *et al.* [28] and Sahin and Unel [22] proposed using ridge regression (RR) regularization in the fitting, which improves (that is, decreases) the condition number of $M^T M$ by adding a term κD to the diagonal of $M^T M$, where κ is a small positive value called the RR parameter, and D is a diagonal matrix. Accordingly Eq. (2) can be modified as

$$(M^T M + \kappa D)\mathbf{a} = M^T\mathbf{b}. \quad (3)$$

See the detail in [28], [22].

D. Shortcomings of Prior Methods

As mentioned in Section I, the first shortcoming of the prior methods is that none of them allow changing the degree during the fitting procedure. Because of this, solving the linear system (2) derived from the least squares method requires that the fixed IP degree must be assigned to construct the inverse of the coefficient matrix $M^T M$. The second shortcoming is related to the local inaccuracy resulting from the conventional RR regularization proposed in [28], [22]. Since the methods cannot rule out the ill condition for the covariant matrix $M^T M$ in time, they have to employ the RR term κD to operate the whole matrix $M^T M$, which leads to losing much local accuracy while global stability is improved. In this paper, we will present a method to solve both these problems.

III. Incremental Fitting

In this section, we present an incremental scheme using QR decomposition for the fitting methods, which allows the IP degree to increase during the fitting procedure until a moderate fitting result is obtained. The computational cost is also 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 needed to increase the IP degree.

A. Fitting using QR decomposition

In place of solving the linear system (2) directly, we first carry out QR decomposition on matrix M as: $M = Q_{N\times m}R_{m\times m}$, where Q satisfies: $Q^T Q = I$ (I is an $m \times m$ identity matrix), and R is an invertible upper triangular matrix.

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

$$\begin{aligned} R^T Q^T QR\mathbf{a} &= R^T Q^T\mathbf{b} \\ \rightarrow R\mathbf{a} &= Q^T\mathbf{b} \\ \rightarrow R\mathbf{a} &= \widetilde{\mathbf{b}}. \end{aligned} \quad (4)$$

After upper triangular matrix R and vector $\widetilde{\mathbf{b}}$ are calculated, the upper triangular linear system can be solved quickly in $\mathcal{O}(m^2)$.

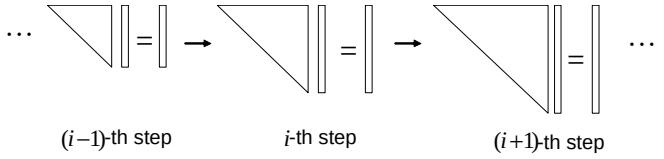

Fig. 2. The incremental scheme that iteratively keeps solving an upper triangular linear system.

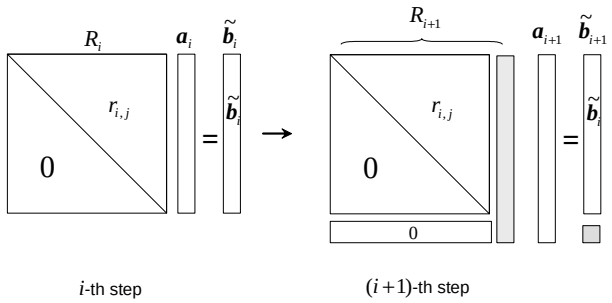

Fig. 3. In order that the triangular linear system grows from the i-th to $(i+1)$-th step, only the calculation shown in light gray is required; other calculation is omitted by reuse at the i-th step.

B. Gram-Schmidt QR Decomposition

Our incremental algorithm depends on the QR-decomposition process. We found that the *Gram-Schmidt* QR Decomposition is very suitable and powerful for saving the computational cost for our algorithm (discussed in the next subsection).

Now, first let us briefly describe the QR decomposition based on the Gram-Schmidt orthogonalization. Assume that matrix M consisting of columns $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_m\}$ is known. The Gram-Schmidt algorithm orthogonalizes $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_m\}$ into the orthonormal vectors $\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m\}$ that are the columns of matrix Q, and simultaneously calculates the corresponding upper triangular matrix R consisting of elements $r_{i,j}$. The algorithm is written in an inductive manner. Initially let $\mathbf{q}_1 = \mathbf{c}_1 / \parallel \mathbf{c}_1 \parallel$ and $r_{1,1} = \parallel \mathbf{c}_1 \parallel$. If $\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_i\}$ have been computed at the i-th step, then the $(i+1)$-th step for orthonormalizing vector $\mathbf{c}_{i+1}$ is

$$
\begin{aligned}
& r_{j,i+1} = \mathbf{q}_j^T \mathbf{c}_{i+1}, \quad \text{for } j \le i, \\
& \mathbf{q}_{i+1} = \mathbf{c}_{i+1} - \sum_{j=1}^{i} r_{j,i+1} \mathbf{q}_j, \\
& r_{i+1,i+1} = \parallel \mathbf{q}_{i+1} \parallel, \\
& \mathbf{q}_{i+1} = \mathbf{q}_{i+1} / \parallel \mathbf{q}_{i+1} \parallel . \qquad (5)
\end{aligned}
$$

With the Gram-Schmidt algorithm, matrix M is successfully QR-decomposed as $M = QR$, and thus the problem of solving Eq. (2) can be transformed to solve a linear system with an upper triangular coefficient matrix.

C. Incremental Scheme

The idea of our incremental scheme is to continuously solve upper triangular linear systems (4) in different dimensions where the QR Decomposition with Gram-Schmidt orthogonalization (5) is utilized. This process is illustrated in Fig. 2, where the dimension of the upper triangular linear system increases, and thus the coefficient vectors of different degrees can be solved.

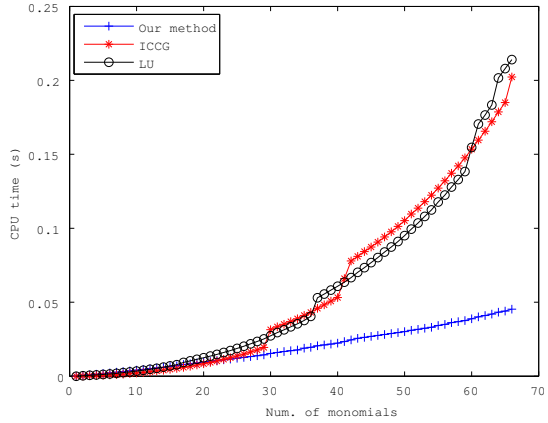

Fig. 4. A comparison between our method and the prior methods with respect to calculation time.

We designed this incremental scheme not only because, at each step, solving an upper triangular linear system is much faster than solving a square one, but also because the calculation for dimension increment between two successive steps is computationally efficient. Fig. 3 illustrates this efficiency by clarifying necessary calculation from the i-th step to the $(i+1)$-th step in our incremental process. For this calculation, in fact, it is only necessary to calculate the parts that are illustrated with gray blocks in Fig. 3.

For constructing the $(i+1)$-th upper triangular linear system from the i-th one, we are concerned with two types of calculation:

1) How to calculate the upper triangular matrix R_{i+1},
2) How to calculate the right-hand vector $\widetilde{\mathbf{b}}_{i+1}$.

Taking advantage of Gram-Schmidt QR Decomposition, we find that, for the first calculation, that is, growing from R_i to R_{i+1}, we only need to calculate the rightmost column of R_{i+1}, and the other elements of R_{i+1} are not changed from R_i; for the second calculation, that is, growing from $\widetilde{\mathbf{b}}_i$ to $\widetilde{\mathbf{b}}_{i+1}$, we only need to calculate the bottom element of $\widetilde{\mathbf{b}}_{i+1}$, and the other elements of $\widetilde{\mathbf{b}}_{i+1}$ are not changed from $\widetilde{\mathbf{b}}_i$.

For the first calculation, the result can be simply obtained from Gram-Schmidt orthogonalization in Eq. (5). For the second calculation, letting $\widetilde{b}_{i+1}$ be the bottom element of vector $\widetilde{\mathbf{b}}_{i+1}$, the calculation of $\widetilde{b}_{i+1}$ can be stated as $\widetilde{b}_{i+1} = \mathbf{q}_{i+1}^T \mathbf{b}$ after carrying out the $(i+1)$-th step of Gram-Schmidt orthogonalization in Eq. (5).

In order to clarify the computational efficiency, let us assume a comparison between our method and a brute-force method, such as the 3L method [1], that iteratively calls the linear method at each step for obtaining the coefficient vectors of different degrees. It is obvious that, for solving coefficient $\mathbf{a}$ at the i-th step, our method needs i inner-product operations for constructing the upper triangular linear system (see (5)), and $\mathcal{O}(i^2)$ for solving this linear system; whereas the latter method needs i^2 inner-product operations for constructing linear system (2), and $\mathcal{O}(i^3)$ for solving Eq. (2).

We show a simple example in Fig. 4 to compare the actual calculation time between our method described above and two other methods that solve the linear system (2) independently at each step with two famous linear system solvers: 1) *LU* decomposition

and 2) incomplete Cholesky conjugate gradient (ICCG). The result was taken from the mean of $10,000$ calculations for solving the same 2D fitting problem that incrementally fits an IP to a certain data set until achieving the 10th degree; practically the IPs of about the 10th degree are often required to represent complex shapes. As shown in this figure, our incremental scheme performs much faster than the other two methods during the dimension-growing process. Note that, although the ICCG algorithm is very efficient for solving a large-scale sparse linear system, matrix M^TM in linear system (2) is often a dense matrix, and therefore ICCG cannot result in good performance.

IV. Global/Local Stabilization

As described in Sections I and II, linear fitting methods usually suffer from the difficulty of achieving global stability. Although ridge regression (RR) regularization can be adopted to achieve global stability [28], [22], local accuracy might deteriorate too much. Addressing this problem, in this section we present a computationally simple procedure to detect whether a specific monomial will contribute to the accuracy of IP representation during the incremental steps. If it will not, RR is used to stabilize the fitting at the current step.

To explain the procedure, we first clarify how RR regularization achieves global stability and why it sacrifices local accuracy. Then we propose a new RR regularization-based stabilization technique: our regularization can concentrate on the improvement of stability by selectively applying RR regularization to our incremental scheme.

A. RR Constraints

First let us analyze why the numerical instability occurs. In fact, an important reason is the collinearity of matrix M, which causes its covariance matrix M^TM to be nearly singular. The collinear columns of M are degenerated to contribute very little to the overall shape of the fit (see [28]). But this little contribution may result in the sensitivity for a high-degree IP, e.g., divergence of some coefficient values. As a result, there are extra undesired solutions generated.

Now let us interpret RR regularization in [4], [28], [22] to be some individual constraints that we call RR constraints hereafter. According to the conventional definition in [4], the formula of RR regularization shown in Eq. (3) can be equivalently transformed as

$$\hat{M}^T\hat{M}\mathbf{a} = \hat{M}^T\hat{\mathbf{b}}, \tag{6}$$

where $\hat{M}$ is the matrix combining matrix M and the square roots of diagonal elements of D, and vector $\hat{\mathbf{b}}$ is from the extension of $\mathbf{b}$ with zeros as follows:

$$\hat{M} = \begin{pmatrix} & M & \\ \sqrt{\kappa d_{11}} & & \\ & \sqrt{\kappa d_{22}} & \\ & & \ddots \\ & & \sqrt{\kappa d_{ll}} \end{pmatrix} \text{ and } \hat{\mathbf{b}} = \begin{pmatrix} \mathbf{b} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where d_{ll} is the l-th diagonal element of D. Eq. (6) is similar to Eq. (2). The only difference between them is that there are some additional row vectors at the bottom of matrix $\hat{M}$, and actually these additional row vectors act as the linear constraints in fitting. Let us call the constraints RR constraints.

These RR constraints overcome the singularity of matrix $\hat{M}^T\hat{M}$ and thus keep the IP presentation globally stable, but an excessive brute-force constraint manner causes all the zero set to be closed to its origin (see [28]), and thus local accuracy deteriorates. Our claim is that if we can apply the RR constraints only to the necessary part, such as the part corresponding to the monomial that weakly contributes to the whole IP representation, the RR regularization can resist deterioration of local accuracy.

B. Proposed RR Regularization

First, it is necessary to detect global instability. Fortunately, in our incremental process, since matrix M has been QR-decomposed as $M = QR$, we can observe that $M^TM = R^TR$, and thus instability of M^TM can be determined from the eigenvalues of R. We can easily evaluate the singularity of R by observing only the diagonal values at each step, since upper triangular matrix R's eigenvalues always lie on its main diagonal.

If the value of the diagonal element r_{ii} at the i-th step is relatively too small, we can assume the current column $\mathbf{c}_i$ of M might be nearly collinear to the previously generated orthogonal space of $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{i-1}\}$ so as to have little contribution to the overall shape of fit. Thus, we apply the i-th RR constraint only to this part to concentrate on solving this collinearity. The action is as follows.

Suppose matrix $\hat{M}$ denotes the matrix M after having added the i-th RR constraint as:

$$\hat{M} = \begin{pmatrix} & M & \\ & & \\ 0 & \cdots & 0 \quad \sqrt{\kappa d_{ii}} \end{pmatrix}.$$

We assume its QR decomposition is $\hat{M} = \hat{Q}\hat{R}$ and each element after the constraint is denoted with $\hat{}$. The difference between M and $\hat{M}$ is only the last row whose last element is only non-zero. From Eq. 5, this effect propagates only r_{ii}, $\mathbf{q}_i$, and $\widetilde{b}_i$. The norm of $\hat{\mathbf{q}}_i$ before normalizing is $\sqrt{r_{ii}^2 + \kappa d_{ii}}$, where r_{ii} is the norm of $\mathbf{q}_i$. Therefore, $\hat{r}_{ii}$ is derived as follows:

$$\hat{r}_{ii} = \sqrt{r_{ii}^2 + \kappa d_{ii}}. \tag{7}$$

From the derivation,

$$\begin{aligned} \hat{b}_i &= \hat{\mathbf{q}}_i^T \begin{pmatrix} \widetilde{\mathbf{b}} \\ 0 \end{pmatrix} \\ &= \frac{r_{ii}}{\sqrt{r_{ii}^2 + \kappa d_{ii}}} \mathbf{q}_i^T \widetilde{\mathbf{b}} = \frac{r_{ii}}{\hat{r}_{ii}} \widetilde{b}_i. \end{aligned} \tag{8}$$

From Eq. (7) we can see obviously that once the i-th eigenvalue r_{ii} is relatively too small, it can be improved to be a larger one as $\hat{r}_{ii}$ by adding the i-th RR constraint. Following this stability improvement, coefficient $\hat{a}_i$ at this step is thus calculated as

$$\hat{a}_i = \frac{\hat{b}_i}{\hat{r}_{ii}} = \frac{r_{ii}}{r_{ii}^2 + \kappa d_{ii}} \widetilde{b}_i \ (< \frac{\widetilde{b}_i}{r_{ii}} = a_i). \tag{9}$$

Also we can see that the divergence of a_i is restrained by adding the i-th RR constraint.

In short, at an incremental step, once a column vector $\mathbf{c}_i$ is detected to be linearly dependent on the preceding columns, the computation in Eqs. (7), (8) and (9) should be done.

C. Normalization for the Linear System

However, checking the value of r_{ii} might be unfair for judging the collinearity. In the result obtained from Gram-Schmidt orthogonalization, r_{ii} might be related not only to the degree of the collinearity but also to the norm of the corresponding column of M. In order to remove only the effect of the norm, it is necessary to normalize the linear system (2).

Normalizing the linear system (2) is performed as follows:

$$M'^T M' \mathbf{a}' = M'^T \mathbf{b}', \tag{10}$$

where M' is the normalized matrix of M and $\mathbf{b}'$ is the normalized vector of $\mathbf{b}$ in Eq. (2) described as

$$M' = \left\{ \frac{\mathbf{c}_1}{\| \mathbf{c}_1 \|}, \frac{\mathbf{c}_2}{\| \mathbf{c}_2 \|}, \ldots, \frac{\mathbf{c}_n}{\| \mathbf{c}_n \|} \right\},$$
$$\mathbf{b}' = \frac{\mathbf{b}}{\| \mathbf{b} \|}, \tag{11}$$

assuming $M = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_n\}$. Therefore the original coefficients can be obtained as $\mathbf{a} = \{\frac{\|\mathbf{b}\|}{\|\mathbf{c}_1\|} a'_1, \frac{\|\mathbf{b}\|}{\|\mathbf{c}_2\|} a'_2, \ldots, \frac{\|\mathbf{b}\|}{\|\mathbf{c}_n\|} a'_n\}$, assuming $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n)$.

Consequently the RR regularization is formulated as

$$(M'^T M' + \kappa D')\mathbf{a}' = M'^T \mathbf{b}'. \tag{12}$$

Here we choose the same selection method for determining RR matrix D' as Tasdizen's method described in [28], [22]. Namely, each diagonal element of D' is chosen as $d'_{ll} = \frac{d_{ll}}{||\mathbf{c}_l||^2}$, where the detail for calculating diagonal elements d_{ll} refers to [28], [22].

An important property of the normalization process is that, if M' is QR-decomposed as $M' = Q'R'$, then all the main diagonal elements of R' vary between 0 and 1, and the elements of vector $Q'^T \mathbf{b}'$ vary between -1 and 1. This beneficial pre-process helps us to make a fair judgment to check out the existence of collinearity in the incremental process, by just checking whether r_{ii} is relatively too small in the range: [0 1].

D. Achieving Euclidean Invariant Method

Unfortunately, the above method is not Euclidean invariant, although it effectively improves the stability for fittings. That might restrain further application to vision applications such as object recognition and pose estimation. For keeping the fitting method Euclidean invariant, the diagonal matrix D in Eq. (3) is designed to satisfy $VDV^T = D$, where V is a block transformation matrix for IP transformation as: $\mathbf{a}' = V\mathbf{a}$ (see [28] and [22]).

According to [28], [22], if we let $D' = KD$, where K is a diagonal matrix and its diagonal elements vary the same as the blocks of V (e.g., $K = diag\{k_0\ k_1\ k_1\ k_2\ k_2\ k_2\ \ldots\ k_i \ldots\ k_n\}$ in 2D), then D' is also Euclidean invariant. The important point is that k_is correspond to elements in the i-th form. Therefore, to extend our method to be a Euclidean invariant version, the incremental fitting procedure can just be modified to increase one-form by one-form, and once the unstable situation is eliminated, then we can improve the eigenvalues corresponding to the whole form.

V. Finding the Moderate Degree

Now the coefficients of various degrees can be worked out stably by the incremental method described above. The remaining problem is how to measure the moderation of degrees for these resolved coefficients. In other words, when should we stop the incremental procedure?

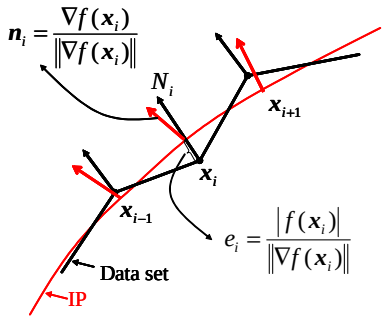

Fig. 5. Definition of two types of our similarity measurement: one is based on the distance between data point and IP zero set, and the other is based on the difference between normal directions associated with data point and IP zero set.

A. Stopping Criterion

For this problem, we first need to define a similarity metric that is capable of measuring the distance between the IP and the data set. Then, based on this similarity metric, we must define a stopping criterion for terminating the incremental procedure. Once this stopping criterion is satisfied, we consider the desired accuracy is reached and the procedure will be terminated.

As shown in Fig. 5, a set of similarity functions can be written as follows:

$$D_{dist} = \frac{1}{N}\sum_{i=1}^{N} e_i, \quad D_{smooth} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{N}_i \cdot \mathbf{n}_i), \tag{13}$$

where N is the number of points, $e_i = \frac{|f(\mathbf{x}_i)|}{\|\bigtriangledown f(\mathbf{x}_i)\|}$, $\mathbf{N}_i$ is the normal vector at a point $\mathbf{x}_i$ obtained from the relations of the neighbors (here we refer to the Sahin's method [22]), and $\mathbf{n}_i = \frac{\bigtriangledown f(\mathbf{x}_i)}{\|\bigtriangledown f(\mathbf{x}_i)\|}$ is the normalized gradient vector of f at $\mathbf{x}_i$. Note e_i is also viewed as the approximation for the Euclidean distance from $\mathbf{x}_i$ to the IP zero set in [29].

D_{dist} and D_{smooth} can be considered as two measurements on distance and smoothness between the data set and the IP zero set. Therefore, the closer D_{dist} and D_{smooth} are to 0 and 1 respectively, the better the fitting become. Then, we can simply define a stopping criterion as

$$(D_{dist} < T_1) \ \wedge \ (D_{smooth} > T_2), \tag{14}$$

where T_1 and T_2 are two tolerances being close to zero and one respectively. For the data set with variation of noise level, an alternative simple way can be considered as:

$$(R_{dist} < T_1) \ \wedge \ (R_{smooth} < T_2). \tag{15}$$

where R_{dist} and R_{smooth} are residuals of D_{dist} and D_{smooth} respectively, namely we only see the difference between the errors of current and previous steps. An example of using this stopping criterion is shown in Section VI E.

B. Algorithm for Finding the Moderate IPs

Given the above conditions, our algorithm can be simply described as follows:

1) Constructing the upper triangular linear system with i column vectors of M' in Eq. (12) with the method described in Section III;

2) Stabilization with the method described in Section IV B;

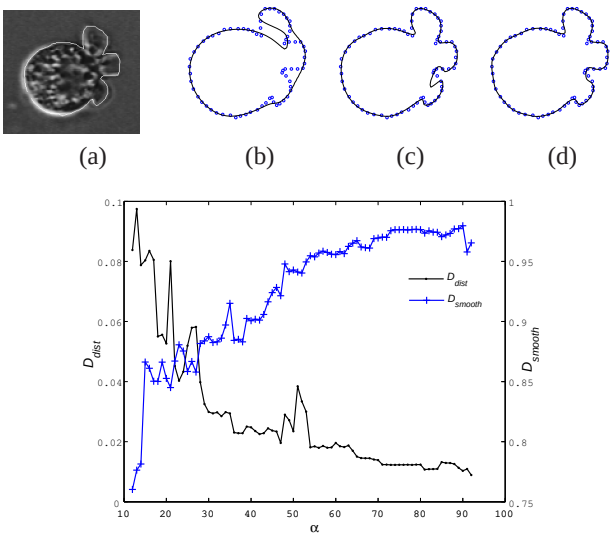

Fig. 6. (a) Original image. IP fitting results at different coefficient numbers: (b) $\alpha = 28$ (six-degree). (c) $\alpha = 54$ (nine-degree). (d) $\alpha = 92$ (thirteen-degree). (e) Graph of D_{dist}. and D_{smooth} vs. coefficient number α. Note "o" symbols represent the boundary points extracted from the image, and solid lines represent the IP zero set in (b)-(d).

3) Solving current linear system to obtain coefficient vector $\mathbf{a}$;
4) Measuring the similarity for the obtained IP;
5) Stopping the algorithm if the stopping criterion (14) or (15) is satisfied; otherwise going back to 1) and increasing the dimension by adding the $(i+1)$-th column of M.

VI. Experimental Results

Our experiments are set in some pre-conditions. 1) As a matter of convenience, we employ the constraints of the 3L method [1] that takes two additional data layers at a distance $\pm\varepsilon$ outside and inside the original data as the optimization constraint[1]. 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 point to origin, as done in [28]; 3) We choose T_1 and T_2 in Eq. (14) with about 0.01 and 0.95 respectively, except for the experiment in Section VI E. 4) The RR parameter κ in Eq. (7) is empirically set to increase the original diagonal element r_{ii} about 10%. Note: we also refer the interested reader to the discussion on setting κ in [28].

A. Examples

In this experiment, we fit an IP to the boundary of a "cell" object shown in Fig. 6(a). The moderate IP is found out automatically based on our stopping criterion (see Fig. 6(d)). To give a comparison, we also show some fits before reaching the desired accuracy (see Fig. 6(b) and (c)). We also show the convergence of D_{dist} and D_{smooth} in Fig. 6(e), which proves the stopping criterion in Eq. (14) can effectively measure the similarity between IPs and data sets.

Some 2-D and 3-D experiments are shown in Fig. 7. The result shows our method's adaptivity for different complexities of shapes.

[1]In our experiment, the layer distance of the 3L method in [1] is set as $\varepsilon = 0.05$.

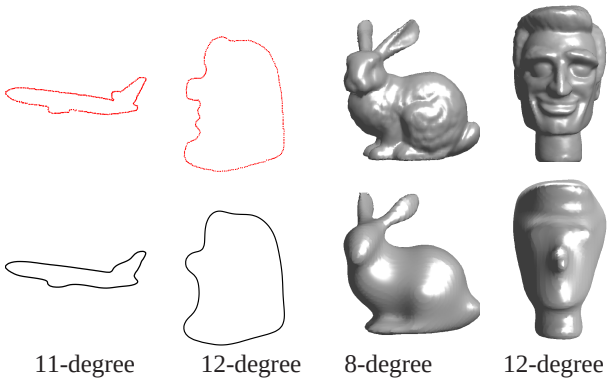

Fig. 7. IP fitting results in 2D/3D. First row: original objects; Second row: IP fits.

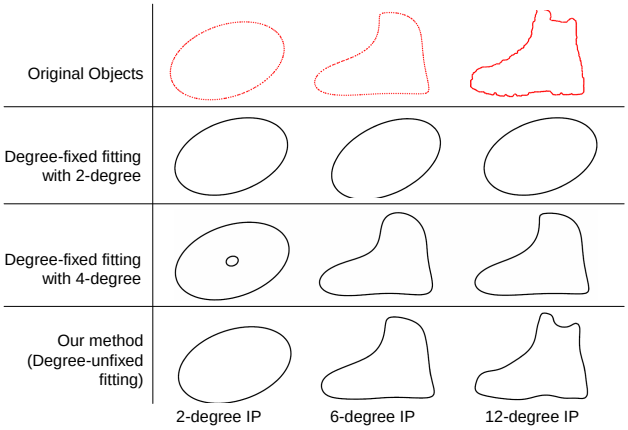

Fig. 8. Comparison between degree-fixed fitting and adaptive fitting. First row: original objects. Second row: IP fits resulting from two-degree degree-fixed fitting. Third row: IP fits resulting from four-degree degree-fixed fitting. Fourth row: IP fits in different degrees resulting from adaptive fitting setting parameters as $T_1 = 0.01$ and $T_2 = 0.95$.

B. Degree-fixed Fitting vs. Adaptive Fitting

Fig. 8 shows some comparisons between the degree-fixed fitting methods and our adaptive fitting method. Compared with degree-fixed methods, in the results of our method, there is neither over-fitting nor insufficient fitting. This 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. To clarify again, as described in Section III-C, our method saves much computational time despite its determination of the moderate degree by an incremental process.

C. Comparison of Fitting Stability

Fig. 9 shows a comparison between three methods: 1) 3L method [1], 2) 3L method [1] + 2D RR regularization [28] and 3L method [1] + 3D RR regularization [22] and 3) our method. As a result, our method shows better performance than the others, in both global stability and local accuracy.

D. An Example for Noisy Data Fitting

We generated some synthetic noisy data sets shown in the top row of Fig. 10. For overcoming the effect on the variation of noise,

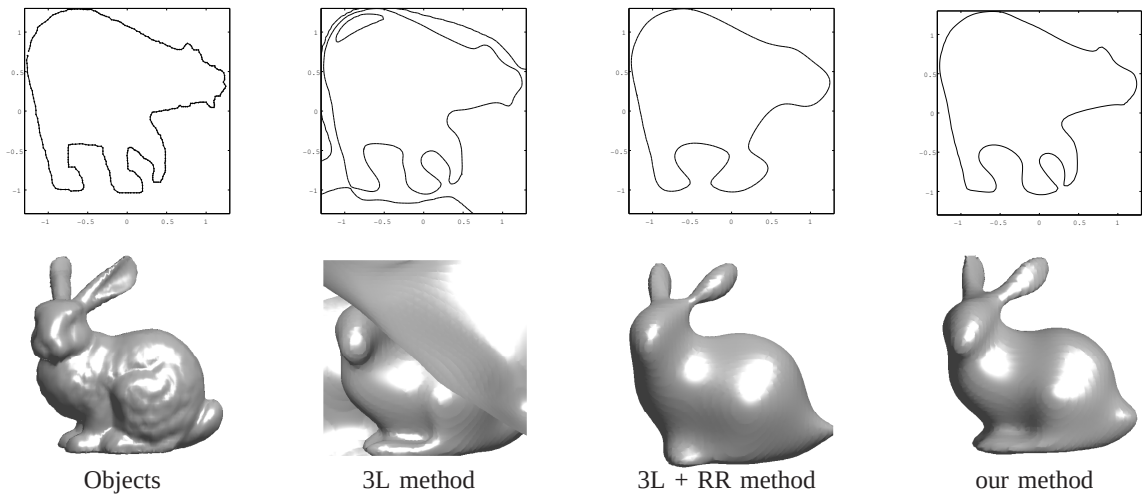

Fig. 9. Comparison of fitting results: first column: original 2D/3D objects, second column: results obtained by 3L method [1], third column: results obtained by 3L method + RR method [28], [22] (for 2D and 3D respectively), last column: results obtained by our method. Results in top row are 12th degree and in bottom row are 8th degree.

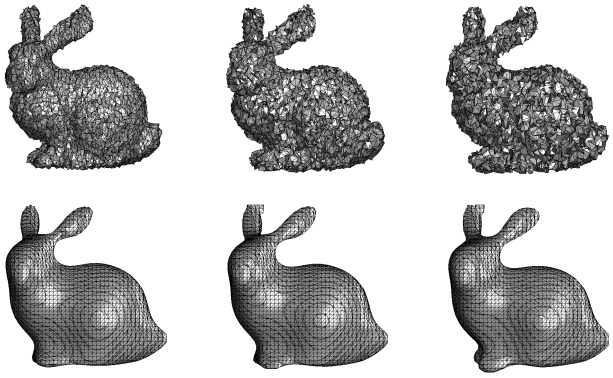

Fig. 10. Top row: Noisy data sets made by adding Gaussian noise to the original model with standard deviations: from left to right 0.1, 0.15, 0.2; Bottom row: the corresponding fitting results.

we adopted the second stopping criterion (15) where parameters with respect to residuals are set as $T_1 = 0.001$ and $T_2 = 0.01$.[2] Then we obtained the fitting results as shown in the bottom row of Fig. 10.

VII. Discussion

A. QR Decomposition Methods

Other famous algorithms of QR decomposition have been presented by Householder and Givens [8]. They have proved more stable than the conventional Gram-Schmidt method in severe cases. But in this paper, since our target is only the regularized data set, we ignore the minor influence of rounding errors. Here we would just like to take advantage of the properties of QR decomposition that orthogonalize vectors one by one, which is the indispensable aspect to achieve our proposed adaptive-degree fitting algorithm.

[2]Since the normals often vary more strongly than vertices, we set the smoothness threshold T_2 to be more tolerant than the distance threshold T_1

B. IP vs Other Functions

In contrast to other function-based representations such as B-spline, NURBS [20], Rational Gaussian [5], radial basis function [32] and topology-preserving polynomial model [11], IP representation cannot give a relatively accurate model. But this representation is more attractive for the applications of fast registration and fast recognition (see the work in [28], [25], [26], [13], [34], [38]), because of its algebraic/geometric invariants [30], [28]. And Sahin and Unel also showed IP's robustness for missing data in [22]. For a more accurate representation of a complex object, it may be required to segment the object shapes and represent each segmented patch with an IP, which also has been considered in our previous work [39]. It is worth noting that another polynomial representation method, topology-preserving polynomial model [11], has the potential of offering an accurate model, but it has not yet been widely used in recognition.

C. Applicability for Recognition

An important advantage of IP for recognition is the existence of algebraic/geometric invariants, which are functions of the polynomial coefficients that do not change after a coordinate transformation. The algebraic invariants that are found by Taubin and Cooper [30], Teral and Cooper [25], Keren [10] and Unsalan [34] are expressed as simple explicit functions of the coefficients. Another set of invariants from the covariant conic decompositions of IP is found by Wolovich *et al.* [36]. A stable coefficient measurement called PIM using IP-interpolated data set is proposed by Lei *et al.* [17].

Our method is able to obtain the coefficients of multiple IP degrees lower than the final fitting degree during the incremental process. Therefore, combining the prior methods described above with these multiple-degree coefficients enables us to extract richer invariant information. Thus it is expected that the invariants have better discriminability; the low-degree coefficients encode shape information roughly, but with less sensitivity to noise, while the high-degree coefficients can encode shape information in detail.

VIII. Conclusions

This paper brings two major contributions. First, it proposes a degree-incremental IP fitting method. The incremental procedure is very fast since the computational cost is saved by the process in which we keep solving upper triangular linear systems of different degrees, and effectively employ the incrementability of the Gram-Schmidt QR decomposition. Thus the method is able to adaptively determine the moderate degree for various shapes.

Second, based on the beneficial property of the upper triangular matrix R in QR decomposition that eigenvalues are the elements on its main diagonal, the fitting instability can be checked out easily and quickly from the diagonal values. By selectively utilizing the RR constraints, our method not only improves global stability, but also maintains local accuracy.

Acknowledgments

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

References

[1] M. Blane, Z. B. Lei, and D. B. Cooper. The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 22(3):298–313, 2000.

[2] A. Fitzgibbon, M. Pilu, and Robert B. Fisher. Direct Least Square Fitting of Ellipses. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 21(5):476–480, 1999.

[3] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant Descriptors for 3D Object Recognition and Pose. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 13(10):971–992, 1991.

[4] G.H. Golub and D. P. O'Leary. Tikhonov Regularization and Total Least Squares. *SIAM J. Matrix Anal. Appl.*, 21:185–194, 2000.

[5] A. Goshtasby. Design and Recovery of 2-D and 3-D Shapes Using Rational Gaussian Curves and Surfaces. *Int. J. Comp. Visi.*, 10(3):233–256, 1993.

[6] A. Helzer, M. Bar-Zohar, and D. Malah. Using Implicit Polynomials for Image Compression. *Proc. 21st IEEE Convention of the Electrical and Electronic Eng*, pages 384–388, 2000.

[7] A. Helzer, M. Barzohar, and D. Malah. Stable Fitting of 2D Curves and 3D Surfaces by Implicit Polynomials. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 26(10):1283–1294, 2004.

[8] R. A. Horn and C. R. Johnson. *Matrix Analysis: Section 2.8*. Cambridge University Press, 1985.

[9] K. Kang, J.-P. Tarel, R. F., and D. Cooper. A Linear Dual-Space Approach to 3D Surface Reconstruction from Occluding Contours using Algebraic Surfaces. *Proc. IEEE Conf. Int. Conf. on Comp. Visi.*, 1:198–204, 2001.

[10] D. Keren. Using symbolic computation to find algebraic invariants. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 16(11):1143–1149, 1994.

[11] D. Keren. Topologically Faithful Fitting of Simple Closed Curves. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 26(1):118–123, 2004.

[12] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 16(1):38–53, 1994.

[13] N. Khan. Silhouette-Based 2D-3D Pose Estimation Using Implicit Algebraic Surfaces. *Master Thesis in Computer Science, Saarland University*, 2007.

[14] K. Knanatani. Renormalization for Computer Vision. *The Institute of Elec., Info. and Comm. eng. (IEICE) Transaction*, 35(2):201–209, 1994.

[15] Z. Lei and D.B. Cooper. New, Faster, More Controlled Fitting of Implicit Polynomial 2D Curves and 3D Surfaces to Data. *Proc. IEEE Conf. Comp. Visi. and Patt. Rec.*, pages 514–522, 1996.

[16] Z. Lei and D.B. Cooper. Linear Programming Fitting of Implicit Polynomials. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 20(2):212–217, 1998.

[17] Z. Lei, T. Tasdizen, and D.B. Cooper. PIMs and Invariant Parts for Shape Recognition. *Proc. IEEE Conf. Int. Conf. on Comp. Visi.*, pages 827–832, 1998.

[18] G. Marola. A Technique for Finding the Symmetry Axes of Implicit Polynomial Curves under Perspective Projection. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 27(3), 2005.

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

[20] L. Piegl. On NURBS: a survey. *Computer Graphics and Applications*, 11(1):55–71, 1991.

[21] N. J. Redding. Implicit Polynomials, Orthogonal Distance Regression, and the Closest Point on a Curve. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 22(2):191–199, 2000.

[22] T. Sahin and M. Unel. Fitting Globally Stabilized Algebraic Surfaces to Range Data. *Proc. IEEE Conf. Int. Conf. on Comp. Visi.*, 2:1083–1088, 2005.

[23] T.W. Sederberg and D.C. Anderson. Implicit Representation of Parametric Curves and Surfaces. *Computer Vision, Graphics, and Image Processing*, 28(1):72–84, 1984.

[24] J. Subrahmonia, D.B. Cooper, and D. Keren. Practical reliable bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 18(5):505–519, 1996.

[25] J. Tarel and D.B. Cooper. The Complex Representation of Algebraic Curves and Its Simple Exploitation for Pose Estimation and Invariant Recognition. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 22(7):663–674, 2000.

[26] J.-P. Tarel, H. Civi, and D. B. Cooper. Pose Estimation of Free-Form 3D Objects without Point Matching using Algebraic Surface Models. *In Proceedings of IEEE Workshop Model Based 3D Image Analysis*, pages 13–21, 1998.

[27] T. Tasdizen and D.B. Cooper. Boundary Estimation from Intensity/Color Images with Algebraic Curve Models. *Int. Conf. Pattern Recognition*, pages 1225–1228, 2000.

[28] T. Tasdizen, J-P. Tarel, and D. B. Cooper. Improving the Stability of Algebraic Curves for Applications. *IEEE Trans. on Imag. Proc.*, 9(3):405–416, 2000.

[29] 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 Patt. Anal. Mach. Intell.*, 13(11):1115–1138, 1991.

[30] G. Taubin and D.B. Cooper. *Symbolic and Numerical Computation for Artificial Intelligence, chapter 6*. Computational Mathematics and Applications. Academic Press, 1992.

[31] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce, and D.J. Kriegman. Parameterized Families of Polynomials for Bounded Algebraic Curve and Surface Fitting. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 16:287–303, 1995.

[32] G. Turk and J. F. OBrieni. Variational Implicit Surfaces. *Technical Report GIT-GVU-99-15, Graph., Visual., and Useab. Cen.*, 1999.

[33] M. Unel and W. A.Wolovich. On the Construction of Complete Sets of Geometric Invariants for Algebraic Curves. *Advances In Applied Mathematics*, 24:65–87, 2000.

[34] C. Unsalan. A Model Based Approach for Pose Estimation and Rotation Invariant Object Matching. *Pattern Recogn. Lett.*, 28(1):49–57, 2007.

[35] C. Unsalan and A. Ercil. A New Robust and Fast Implicit Polynomial Fitting Technique. *Proc. M2VIP 99*, 1999.

[36] W. A. Wolovich and M. Unel. The Determination of Implicit Polynomial Canonical Curves. *IEEE Trans. on Patt. Anal. Mach. Intell.*, 20(10):1080–1090, 1998.

[37] H. Yalcin, M. Unel, and W. Wolovich. Implicitization of Parametric Curves by Matrix Annihilation. *Int. J. Comp. Visi.*, 54(1/2/3):105–115, 2003.

[38] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, and K. Ikeuchi. 6-DOF Pose Estimation from Single Ultrasound Image using 3D IP Models. *Proc. IEEE Conf. Computer Vision and Patt. Rec. (CVPR) Workshop on Ojb. Trac. Class. Beyond Visi. Spec. (OTCBVS'08)*, 2008.

[39] B. Zheng, J. Takamatsu, and K. Ikeuchi. 3D Model Segmentation and Representation with Implicit Polynomials. *To appear in IEICE Trans. on Information and Systems*, E91-D(4), 2008.