Evolving Implicit Polynomial Interfaces

Paper xxx

Abstract

Although algebraic or so called "implicit polynomial" curves/surfaces have been studied rather extensively for several decades, to the best of our knowledge, a dynamic formulation of them, similar to active contours, has not been done yet. This paper proposes to use implicit polynomial functions instead of signed distance functions in classical level set formulation and shows potential of such representations. In particular, it is shown that utilization of an implicit polynomial distance function in the level set equation yields an ordinary differential equation (ODE) for the temporal behavior of the polynomial coefficients. Using a control theoretic approach, several problems such as curve morphing, dynamic conic fitting without and with constraint, i.e. dynamic ellipse fit, and dynamic curve fitting can be tackled within this new framework. Results are verified by several examples on real images.

1 Introduction

To model and track evolving interfaces is an important problem in several disciplines including vision and computational physics. Since the seminal work of Kass et al. [1], several active contour paradigms have been developed [2]-[4]. Much recent work has focused on evolving interfaces using a PDE based approach, i.e. level set formulation [5]-[7]. Applications include image restoration, segmentation, image inpainting and image classification.

Algebraic curves and surfaces have been used in various branches of engineering for a long time, but in the past two decades they have proven very useful in many model-based applications. Various algebraic and geometric invariants obtained from implicit models of curves and surfaces have been studied rather extensively in computer vision, especially for single computation pose estimation, shape tracking, 3D surface estimation from multiple images and efficient geometric indexing of large pictorial databases [8]- [11].

Although there has been extensive work on algebraic curves and surfaces, to the best of our knowledge, there is no work on evolving algebraic curves/surfaces. This paper, we hope, would perhaps initiate a strategy for evolving implicit polynomial curves/surfaces.

In this work, we propose to use implicit polynomial level set functions instead of signed distance functions in the classical level set formulation and show potential of such representations. Utilization of an implicit polynomial function in the level set equation results in an ordinary differential equation (ODE) for the temporal behavior of the polynomial coefficients. Using a control theoretic approach, we will show how to morph a curve, how to develop a dynamic conic fitting algorithm without constraint and with constraint, i.e. dynamic ellipse fit, and how to develop a dynamic fitting method for higher degree curves.

Organization of this paper is as follows: In Section 2, we present evolving implicit polynomial interfaces using the level set equation and obtain an ordinary differential equation (ODE) for the temporal behavior of the coefficient vector of the polynomial. In Section 3, we outline a new procedure for curve morphing and provide several examples. In Section 4, we introduce a dynamic conic fitting method without and with constraint. Experimental results on real images are also given. In section 5, a dynamic fitting technique for higher degree curves is developed and verified by an example. Section 6 concludes the paper with some remarks and shows future directions.

2 Evolving Implicit Polynomial Interfaces

We are interested in how implicit polynomials curves/surfaces can be used in the level set framework [6]. The level set equation is given by the following Partial Differential Equation (PDE):

$$\Phi_t + \nabla\Phi^T V = 0 \tag{1}$$

where Φ is the level set function and $V = (V_x, V_y)$ is the velocity of the interface. In the literature, Φ is usually chosen as a signed-distance function [6]. In this work, we will consider a degree d bivariate polynomial with time varying coefficients, namely

$$\Phi(x,y,t) = \sum_{0\le i,j;i+j\le d} \alpha_{ij}(t)x^i y^j$$

$$= \underbrace{\begin{pmatrix} x^d & x^{d-1}y & \dots & 1 \end{pmatrix}}_{m^T} \underbrace{\begin{pmatrix} \alpha_{d0} & \alpha_{d-1,1} & \dots & \alpha_{00} \end{pmatrix}^T}_{\alpha} = m^T\alpha \tag{2}$$

where m is the vector of monomials and α is the coefficient vector. Substitution of (2) into (1) implies

$$m^T\dot{\alpha} + (V_x m_x^T + V_y m_y^T)\alpha = 0 \tag{3}$$

or,

$$m^T\dot{\alpha} + m_g^T\alpha = 0 \tag{4}$$

where m_x and m_y are the partial derivatives of the monomial vector m with respect to x and y, and $m_g = V_x m_x + V_y m_y$. Writing (4) for $N \ge c$ interface points, we obtain

$$M\dot{\alpha} + M_g\alpha = 0 \tag{5}$$

where $c = (d+1)(d+2)/2$ is the size of the coefficient vector α, M is a $N \times c$ matrix obtained by stacking vectors of monomials (m), and M_g is a $N \times c$ matrix of partial derivatives obtained by stacking m_g vectors. In practice, (5) will be satisfied only approximately, and therefore an Ordinary Differential Equation (ODE) which governs the temporal behavior of the coefficient vector can be obtained as

$$\dot{\alpha} = -M^\dagger M_g\alpha \tag{6}$$

where $M^\dagger = (M^T M)^{-1}M^T$ is the pseudo-inverse of M.

In light of (6), a discrete update rule for the coefficient vector can be obtained as follows:

$$\alpha_{k+1} = (I - \Delta t M^\dagger M_g)\alpha_k \tag{7}$$

where Δt is the sampling time.

3 Curve Morphing

Metamorphosis of curves can be defined as the gradual transformation of one curve into another over time [12] and has an important role in many applications such as geometric modelling [12], industrial design [13], creation of visual effects and computer animations, etc. Here we metamorphose an initial curve of degree d_1 to a final reference curve of degree d_2,

$$f_{d_1}(x,y) \longrightarrow f^*_{d_2}(x,y)\ , \qquad d_1 \leq d_2 \tag{8}$$

where $f_{d_1}(x,y)$ and $f^*_{d_2}(x,y)$ have $c_1 = (d_1+1)(d_1+2)/2$ and $c_2 = (d_2+1)(d_2+2)/2$ coefficients, respectively. For $d_1 < d_2$, the number of coefficients of $f_{d_1}(x,y)$ will be $c_2 - c_1$ less than that of $f^*_{d_2}(x,y)$. Let α and α^* be the coefficient vectors of $f_{d_1}(x,y)$ and $f^*_{d_2}(x,y)$, respectively. Since α and α^* have different sizes, we introduce the following augmented coefficient vector for $f_{d_1}(x,y)$,

$$\bar{\alpha} = \begin{pmatrix} \varepsilon & \dots & \varepsilon & \alpha^T \end{pmatrix}^T \tag{9}$$

where α is preceded with $c_2 - c_1$ extremely small positive numbers $\varepsilon = 10^{-c_2}$.

In order to achieve metamorphosis, we define an error function in terms of the coefficient vectors

$$e = \alpha^* - \bar{\alpha} \tag{10}$$

By differentiating (10), we get

$$\dot{e} = -\dot{\bar{\alpha}} \tag{11}$$

In light of (6), $\dot{\bar{\alpha}}$ can be written as

$$\dot{\bar{\alpha}} = -M^{\dagger} M_g \bar{\alpha} = -M^{\dagger} u \tag{12}$$

where $u \stackrel{\text{def}}{=} M_g \bar{\alpha}$. Recall that M_g was a function of the velocity of the interface, and therefore we will treat u as a control variable. By imposing $\dot{e} = -\Lambda e$, an exponential decrease of the error can be realized by selecting the control variable according to

$$-\Lambda e = -(-M^{\dagger} u) \tag{13}$$

where $\Lambda \in \Re^{c_2 \times c_2}$ is a positive-definite gain matrix with the diagonal elements $\lambda_i > 0$ for $i = 1,2,\dots,c_2$. Rearranging (13) we obtain the control variable as

$$u = -(M^{\dagger})^{\dagger} \Lambda e \tag{14}$$

3.1 Examples

Figure 1 depicts the evolution of a circle (f_2) towards an ellipse (f^*_2) whose coefficient vector is

$$\alpha^* = \begin{bmatrix} 1 & -1.2 & 1 & 0 & 0 & -1.6 \end{bmatrix}^T$$

Figure 2 shows convergence of the error and of the coefficients. The value of the α at 216^{th} iteration is

$$\alpha = \begin{bmatrix} 1.0000 & -1.1994 & 1.0000 & 0.0009 & 0.0008 & -1.5985 \end{bmatrix}^T$$

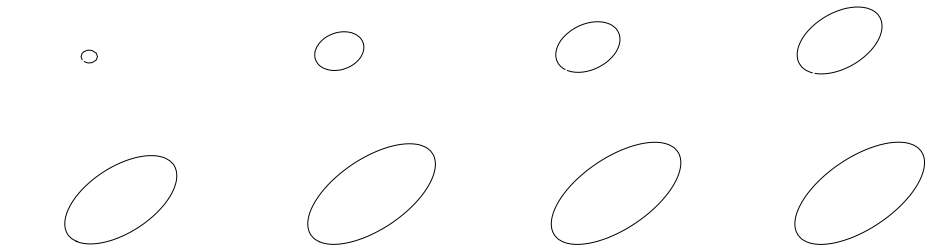
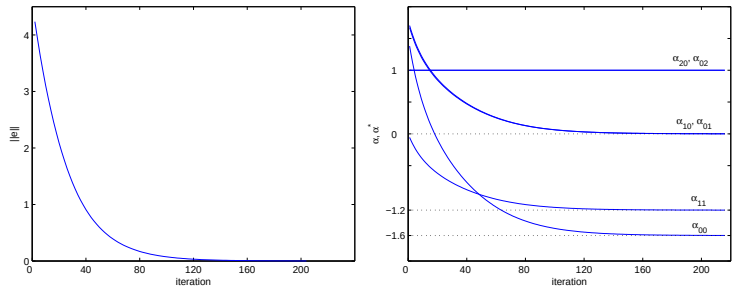


Figure 1: Transition of a circle into an ellipse.



(a) (b)

Figure 2: (a) Exponential decrease of the error norm (b) Convergence of α towards α^*

Figure 3 depicts the metamorphosis of a circle (f_2) into an aircraft shape (f_6^*). The coefficient vector of f_2 is defined as

$$\alpha = \begin{bmatrix} 1 & 0 & 1 & -2x_c & -2y_c & (x_c^2 + y_c^2 - r^2) \end{bmatrix}^T$$

where (x_c, y_c) and r are the center and radius of initial circle. Similarly, the coefficient vector of f_6^* is given as

$$\alpha^* = [1 \ -0.407 \ -2.437 \ 1.226 \ 6.958 \ -0.451 \ 0.465 \ 3.762 \ -1.642 \ -10.212\ldots$$
$$0.767 \ -5.102 \ 0.209 \ 1.875 \ 0.321 \ 6.973 \ -0.842 \ -1.424 \ -6.505 \ 1.281\ldots$$
$$4.846 \ -0.158 \ -4.238 \ 0.25 \ 0.539 \ 3.908 \ -0.236 \ -1.014]^T$$

Figure 4 shows convergence of the error norm for two different values of λ_i where $i = 1, 2, \ldots, 28$.

Remark We can increase or decrease the speed of transition and/or convergence by playing with proportional gains, λ_i. Exponential decrease of the error can also be achieved by nonlinear control laws.

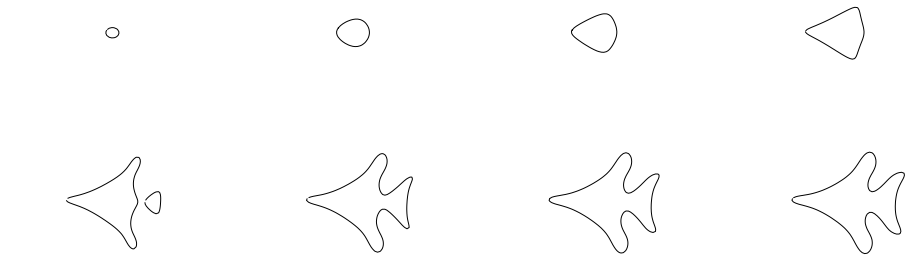
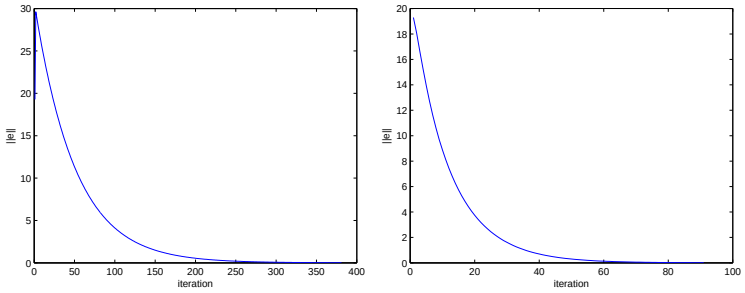


Figure 3: Transition of a circle into an aircraft ($\lambda_i = 2$)



(a) (b)

Figure 4: Convergence of the error norm for $\lambda_i = 2$ (a) and for $\lambda_i = 8$ (b)

4 Dynamic Conic Fitting

We can fit an evolving conic $f_2(x,y)$ to an edge contour data $X^* = \{(x_i,y_i)|_{i=1}^N\}$ by minimizing the following error function

$$e = f_2(X^*) = M^*\alpha$$

$$e = \underbrace{\begin{pmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ & & \vdots & & & \\ x_N^2 & x_Ny_N & y_N^2 & x_N & y_N & 1 \end{pmatrix}}_{M^*} \underbrace{\begin{pmatrix} \alpha_{20} \\ \alpha_{11} \\ \alpha_{02} \\ \alpha_{10} \\ \alpha_{01} \\ \alpha_{00} \end{pmatrix}}_{\alpha} \tag{15}$$

where $M^* \in \Re^{N\times 6}$ is the monomial matrix created from X^* and $\alpha \in \Re^6$ is the coefficient vector of the curve $f_2(x,y)$ which will evolve towards target data. In order to obtain the

error dynamics, equation (15) is differentiated with respect to time, which yields

$$\dot{e} = M^* \dot{\alpha} \tag{16}$$

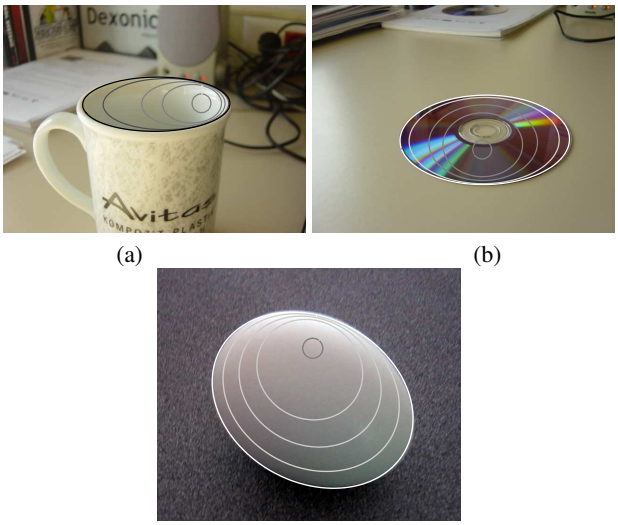
Imposing $\dot{e} = -\Lambda e$ and substituting (12) into (16), we obtain

$$-\Lambda e = -M^* M^\dagger u \tag{17}$$

where $\Lambda \in \Re^{N \times N}$ is a positive-definite diagonal gain matrix with diagonal elements $\lambda_i > 0$ for $i = 1, 2, .., N$. The control variable u can be derived from (17) as

$$u = (M^* M^\dagger)^\dagger \Lambda e \tag{18}$$

Figure 5 shows results of dynamic conic fitting on real images of a cup-mouth, a DVD, and an egg along with some of their evolution curves. The initial positions of the curves are marked by the user with a mouse. The edge contours of the objects are extracted using a simple edge detection algorithm.



(a) (b)

(c)

Figure 5: Dynamic conic fitting on real images. Curves are evolving towards the boundaries of a cup-mouth (a), a DVD (b), and an egg (c)

4.1 Dynamic Conic Fitting with Constraint

In many cases, it may not be possible to obtain a sharp edge contour of the object due to partial occlusions, noise from the vision sensors etc. The previous conic fitting algorithm may not cope with such circumstances. In order to impose ellipse constraint in the fitting process, we append the constraint $\alpha_{11}^2 - 4\alpha_{20}\alpha_{02} = -1$, as proposed in [14], to the former algorithm and rewrite the error function as

$$\begin{aligned} e_1 &= M^* \alpha \\ e_2 &= \alpha^T C \alpha + 1 \end{aligned} \tag{19}$$

where $M^* \in \Re^{N\times 6}$ is the monomial matrix as in (15) and $C \in \Re^{6\times 6}$ is the constraint matrix defined as

$$C = \begin{bmatrix} 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

to satisfy $\alpha_{11}^2 - 4\alpha_{20}\alpha_{02} + 1 = 0$ in e_2 of (19). The differentiation of the error system yields,

$$\begin{aligned} \dot{e}_1 &= M^*\dot{\alpha} \\ \dot{e}_2 &= 2\alpha^T C\dot{\alpha} \end{aligned} \tag{21}$$

which can be rewritten in matrix-vector form as

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} M^* \\ 2\alpha^T C \end{bmatrix}}_{S} \dot{\alpha} \tag{22}$$

where $S \in \Re^{(N+1)\times 6}$ is the new design matrix which is combination of the monomial and the constraint matrices. Substitution of (12) into (22) implies

$$\dot{e} = S\dot{\alpha} = -SM^{\dagger}u \tag{23}$$

Imposing $\dot{e} = -\Lambda e$ for the exponential decrease of the error, we get

$$-\Lambda e = -SM^{\dagger}u \tag{24}$$

where $\Lambda \in \Re^{(N+1)\times(N+1)}$ is a diagonal gain matrix with the elements $\lambda_i|_{i=1}^{N} > 0$ and $\lambda_{N+1} > 0$ to control the regulation of algebraic distances and the constraint, respectively. Control variable u can be obtained by solving (24) as

$$u = (SM^{\dagger})^{\dagger}\Lambda e \tag{25}$$

We now present some examples of this constrained dynamic conic fitting method. Figures 6 and 7 depict the results for an egg and a mouse contours with missing data due to partial occlusion.

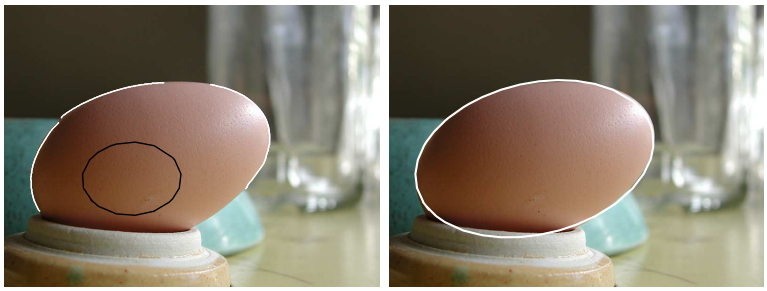
5 Dynamic Fitting of Higher Degree Implicit Curves

In this section, we will show how to evolve higher degree implicit curves for dynamic fitting of complicated objects by utilizing the 3L curve fitting approach [15].

5.1 Dynamic 3L Fitting

Let $L^- = \{(x,y) : f_d(x,y) = -l\}$ and $L^+ = \{(x,y) : f_d(x,y) = +l\}$ be the inner and the outer level sets for a degree d implicit polynomial curve $f_d(X^*) = 0$ with X^* being the edge contour of the object. The following errors can be defined

$$\begin{aligned} e_1 &= M_{-}^{*}\alpha + \bar{l} \\ e_2 &= M^*\alpha \\ e_3 &= M_{+}^{*}\alpha - \bar{l} \end{aligned} \tag{26}$$



(a) (b)

Figure 6: (a) An egg contour with missing data (white) and initial curve (black), (b) result of dynamic ellipse fit on the egg.

(a) (b)

Figure 7: (a) A mouse contour and the initial curve, (b) result of dynamic ellipse fit.

where $M_{-}^{*} \in \Re^{N\times c}$ and $M_{+}^{*} \in \Re^{N\times c}$ are the level set monomial matrices of L^{-} and L^{+}, respectively, and $\bar{l} = [l,\ l,\ \ldots,\ l]^{T} \in \Re^{N}$ is the distance vector for level sets. Differentiating the errors with respect to time yields

$$\begin{aligned} \dot{e}_1 &= M_{-}^{*}\dot{\alpha} \\ \dot{e}_2 &= M^{*}\dot{\alpha} \\ \dot{e}_3 &= M_{+}^{*}\dot{\alpha} \end{aligned} \tag{27}$$

$$\dot{e} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} M_{-}^{*} \\ M^{*} \\ M_{+}^{*} \end{bmatrix}}_{S} \dot{\alpha} \tag{28}$$

Using (12) and imposing $\dot{e} = -\Lambda e$, it follows that

$$\dot{e} = S\dot{\alpha} = -SM^{\dagger}u \tag{29}$$

$$-\Lambda e = -SM^{\dagger}u \tag{30}$$

where $\Lambda \in \Re^{3N\times 3N}$ is a positive-definite diagonal gain matrix constructed as

$$\Lambda = diag(\lambda_1^{-},\ldots,\lambda_N^{-},\lambda_1,\ldots,\lambda_N,\lambda_1^{+},\ldots,\lambda_N^{+}) \tag{31}$$

Solution of (30) for the control variable u can be obtained as

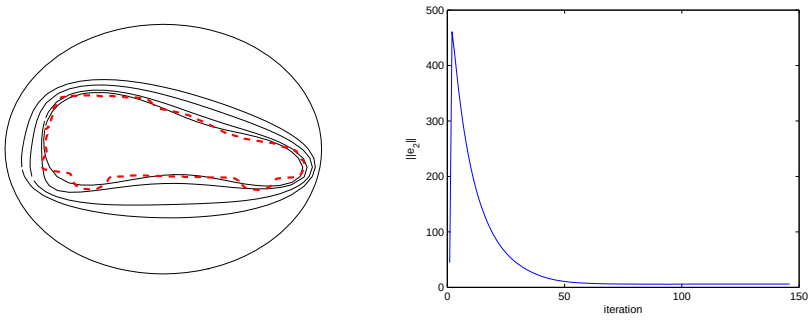
$$u = (SM^{\dagger})^{\dagger}\Lambda e \tag{32}$$

Figure 8 depicts the evolution of a circle into a van using dynamic 3L fitting. Initial curve (circle) is defined by

$$\alpha = [\varepsilon, \ldots, \varepsilon, 1, 0, 1, -2x_c, -2y_c, (x_c^2 + y_c^2 - r^2)]^T \in \Re^{15}$$

where $r = 1.8,\ \lambda^{\mp} = 6,\ \lambda = 4,\ \varepsilon = 10^{-4}$. Final curve is obtained as

$$\alpha = [1, -0.2779, 0.857, -0.8002, 2.8879, -0.8112, 1.9849, 2.3037, \ldots$$
$$-0.0116, -1.1576, 2.8479, 5.2185, 1.3863, -1.3331, -1.3405]^T$$



(a) (b)

Figure 8: (a) Evolution of a circle into a van, (b) error graph of $||e_2||$

6 Conclusion and Future Work

We have now formulated evolving implicit polynomial curves/surfaces. Specifically, we have proposed to use implicit polynomial level set functions instead of signed distance functions in the classical level set formulation. Utilization of an implicit polynomial function in the level set equation resulted in an ordinary differential equation (ODE) which governs the temporal behavior of the polynomial coefficients. Introducing a control theoretic approach, we have shown how to morph a curve, how to develop a dynamic conic fitting algorithm without constraint and with constraint, i.e. dynamic ellipse fit, and how to develop a dynamic fitting method for higher degree curves.

As an extension, we are currently working on a full segmentation problem in which one initiates an arbitrary closed-bounded algebraic curve, i.e. a circle or an ellipse, and updates its coefficients according to the ODE mentioned above, and finally stops it on the object boundary. Much work has to be done in this direction.

References

[1] M. Kass, A. Witkin, and D. Terzopoulus, Snakes: Active Contour Model, Int'l J. Computer Vision, vol. 1, no. 4, pp. 321-331, 1988.

[2] V. Caselles, R. Kimmel, and G. Sapiro, Geodesic Active Contour, Int'l J. Computer Vision, vol. 22, no. 1, pp. 61-79, 1997.

[3] C. Xu, J.L. Prince, Gradient vector flow: A new external force for snakes, IEEE Proc. Conf. on Comput. Vis. Patt. Recog. (CVPR), pp. 66-71, 1997.

[4] C. Xu and J. Prince, Generalized Gradient Vector Flow External Forces for Active Contours, Signal Processing, vol. 71, no. 2, pp. 131-139, 1998.

[5] J. Sethian, Level Set Methods, Cambridge Univ. Press, 1996.

[6] S. Osher and J. Sethian, Fronts Propagating with Curvature Dependent Speed, Algorithms Based on a Hamilton-Jacobi Formulation, J. Comput. Phys., 79:12-49, 1988.

[7] N. Paragios and R. Deriche, Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation, Int'l J. Computer Vision, vol. 46, no. 3, pp. 223-247, 2002.

[8] D. Keren et al., "Fitting curves and surfaces to data using constrained implicit polynomials," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 1, January 1999.

[9] M. Unel, W. A. Wolovich, "A new representation for quartic curves and complete sets of geometric invariants," International Journal of Pattern Recognition and Artificial Intelligence, December 1999.

[10] G. Taubin, D. B. Cooper, "2D and 3D object recognition and positioning with algebraic invariants and covariants," Chapter 6 of **Symbolic and Numerical Computation for Artificial Intelligence**, Academic Press, 1992.

[11] G. Taubin, F. Cukierman, S. Sullivan, J. Ponce and D.J. Kriegman, "Parameterized families of polynomials for bounded algebraic curve and surface fitting," IEEE PAMI, March, 1994.

[12] L. Liu, G. Wang, B. Zhang, B. Guo and H. Shum, Perceptually Based Approach For Planar Shape Morphing, Computer Graphics and Applications, 111-120, 2004.

[13] K. C. Hui and Y. Li, Feature-Based Shape Blending Technique For Industrial Design, Computer-Aided Design, 30(10), 823-834, 1998.

[14] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher, Direct Least Square Fitting of Ellipses, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, MAY 1999.

[15] Z. Lei, M.M. Blane, and D.B. Cooper, 3L Fitting of Higher Degree Implicit Polynomials, Proc. Third IEEE Workshop Applications of Computer Vision, pp. 148-153, Dec. 1996.