International Journal of Computer Vision 54(1/2/3), 105–115, 2003

Implicitization of Parametric Curves by Matrix Annihilation*

HULYA YALCIN
Division of Engineering, Brown University, Providence, RI, USA
hulya_yalcin@brown.edu

MUSTAFA UNEL
Center for Computational Vision and Control, Yale University, CT, USA
mu@sysc.eng.yale.edu

WILLIAM WOLOVICH
Division of Engineering, Brown University, Providence, RI, USA
waw@lems.brown.edu

Received April 25, 2001; Revised February 11, 2002; Accepted July 25, 2002

Abstract. Object recognition is a central problem in computer vision. When objects are defined by boundary curves, they can be represented either explicitly or implicitly. Implicit polynomial (IP) equations have long been known to offer certain advantages over more traditional parametric methods. However, the lack of general procedures for obtaining IP models of higher degree has prevented their general use in many practical applications. In most cases today, parametric equations are used to model curves and surfaces. One such parametric representation, elliptic Fourier Descriptors (EFD), has been widely used to represent 2D and 3D curves, as well as 3D surfaces. Although EFDs can represent nearly all curves, it is often convenient to have an implicit algebraic description $F(x, y) = 0$, for several reasons. Algebraic curves and surfaces have proven very useful in many model-based applications. Various algebraic and geometric invariants obtained from these implicit models have been studied rather extensively, since implicit polynomials are well-suited to computer vision tasks, especially for single computation pose estimation, shape tracking, 3D surface estimation from multiple images and efficient geometric indexing into large pictorial databases. In this paper, we present a new non-symbolic implicitization technique called the *matrix annihilation method*, for converting parametric Fourier representations to algebraic (implicit polynomial) representations, thereby benefiting from the features of both.

Keywords: free-form models, implicitization, Fourier descriptors, implicit polynomials

1. Introduction

Object recognition and pose estimation in computer vision provides a sizeable literature on alignment and invariants based on moments, B-splines, superquadrics, conics, differential invariants and Fourier descriptors (Taubin, 1991; Huang and Cohen, 1996; Solina and Bajcsy, 1990; Ma, 1993; Mundy and Zisserman, 1992; Calabi et al., 1998) Shape representation based on parametric representations has been studied extensively, and the use of parametric representations remains dominant in computer graphics, computer vision and geometric modeling. However, algebraic models have certain mathematical and computational advantages complementary to the parametric methods, and they are receiving increased attention.

*This work was supported by the NSF under Grant DMI-9820804.

Algebraic curves/surfaces have proved to be very useful for shape representation (Taubin et al., 1994; Keren et al., 1994; Bloomenthal, 1997). Invariants associated with algebraic models have also been employed in several model-based vision and pattern recognition applications (Wolovich and Unel, 1998; Unel and Wolovich, 1998, 1999, 2000; Subrahmonia et al., 1996). In the past few years, implicit representations have been used more frequently, allowing a better treatment of several problems. It is sometimes more convenient to have an implicit equation in applications such as determining curve/surface intersections and the point classification problem, since they imply a simple evaluation of the implicit functions. Implicit polynomials are also well-suited for determining "how close" measured points on a curve/surface are to the ideal curve/surface, once the ideal surface is modeled with an implicit polynomial (Wolovich et al., 2002).

There is very little in the literature on higher degree IP models for large or entire free-form shapes because of the lack of tractable computational procedures for obtaining and analyzing such models. The problem of excessive number of parameters in implicit representations was first studied in Subrahmonia et al. (1996). The implicit polynomial that defines a curve is not easily determined. Several different fitting algorithms have been proposed for directly determining IP models from point data sets. The linear 3L fitting algorithm (Lei et al., 1996) often exhibits very fast and accurate curve representation and stability. Continuous improvements are being made on the stability of algebraic curve fitting for obtaining IP models (Tasdizen et al., 2000). However, there is significant value in determining other IP methods.

Instead of obtaining algebraic curve representations directly from points, it is also possible to convert parametric EFD equations (which may be obtained from boundary points) to implicit ones, which the main focus here. The general process of converting from parametric equations of curves or surfaces to implicit ones is known as *implicitization*, and it has been studied for more than a century.

Salmon performed surface implicitization by eliminating parameters from the parametric surface equations (Salmon, 1915). In 1908, Dixon published a more compact resultant for eliminating two variables from three polynomials, which became the standard method for surface implicitization in the absence of base points (Dixon, 1908). Implicitization techniques based on elimination theory also have been extensively used by Sederberg (1983), Sederberg and Anderson (1984), and Sederberg and Goldman (1986) and involve computing the determinant of Sylvester's matrix. A significant disadvantage of implicitization by Sylvester's matrix elimination method is that it involves computing the determinant of a matrix which contains symbolic variables. Using Dixon's formulation, Sederberg implicitizes tensor product surfaces (Sederberg, 1983). Recently, Sederberg et al. introduced an implicitization method called the moving quadric method (Sederberg and Chen, 1995) where they lower the size of the matrix from which the resultant is formed by moving curves.

Hong employed Sylvester's matrix for a particular class of problems involving trigonometric polynomials, taking advantage of their special structure (Hong, 1995). Hobby (1991) used the same general approach as Sederberg, but introduced singular value decomposition and rotated coordinates to enhance the numerical stability of implicitization for polynomial cubic curves (Hobby, 1991). Macaulay's (1916) formulation expresses the resultant as a ratio of two determinants. Bajaj et al. (1988) and Chionh (1990) use such procedures to compute the resultant of three parametric equations for implicitizing. In Canny (1990), Canny computed the resultant by perturbing the equations when determinants were zero, which introduces an additional variable and increases the symbolic complexity of the resulting expression. As noted in Hoffmann (1989), many techniques based on elimination theory can result in extraneous factors along with the implicit equation, and separating them can be a time consuming task.

Unsalan and Ercil (2001) studied the problem of converting between parametric and implicit forms based on polar/spherical coordinate representations. However, their technique is valid only for star-shaped curves. Many of these and other methods of implicitization, such as multivariate resultants (Chionh and Goldman, 1992) are surveyed in Hoffmann (1993).

Another technique for implicitization utilizes Groebner bases by computing a canonical representation of the ideal generated by the parametric equations by defining a suitable ordering of the variables (Buchberger, 1985, 1989). The problem of implicitizing parametric surfaces without any base points can be done using Groebner bases or resultants as shown in Buchberger (1989) and Manocha and Canny (1992b), although it is fairly complex in practice. Several other procedures have been devised to implicitize surfaces with base points (Hoffmann, 1989; Chionh, 1990; Manocha and Canny, 1992a, 1992b).

Here, we note that any closed curve can be described in terms of a set of two (or three for space curve) Fourier series whose coefficients are called elliptic Fourier descriptors (EFDs). The advantages of using EFDs are that the shape information is concentrated in the low frequency parts (Wallace and Mitchell, 1980; Kuhl and Giardina, 1982; Zahn and Roskies, 1972; Granlund, 1972; Lin and Hwang, 1987), so that shape can be described by the first few coefficients. Invariants derived from EFDs have been used for identification (Lin and Hwang, 1987; Lestrel, 1997). Recently, Sheu and Wu have proposed a scheme to obtain two-variable 3D FDs to describe both axisymmetric and nonaxisymmetric objects, which are generated by computing 2D FDs from 2D coordinates followed by an iterative computation of 3D FDs (Wu and Sheu, 1998).

In this paper, we present a new non-symbolic implicitization technique, called the *matrix annihilation* method, for converting parametric EFD representations to algebraic ones. We should note that our method is numerical, so that we can directly parametrize more complex curves with higher order polynomial degrees. Furthermore, our procedure is computationally efficient.

The structure of the paper is as follows: Section 2 reviews implicitization by Sylvester's matrix. In Sections 3 and 4, our approach for implicitization of closed curves is explained and illustrated by examples. Section 5 describes some vision-based examples and some concluding remarks are given in Section 6.

2. Review of Implicitization by Sylvester's Matrix

Sylvester's matrix elimination method (Sederberg, 1983; Hoffmann, 1989) can be used to implicitize parametric polynomial curves such as

$$x = a_2t^2 + a_1t + a_0 \quad y = b_2t^2 + b_1t + b_0$$

In particular, if we re-write these equations as

$$a_2t^2 + a_1t + (a_0 - x) = 0 \quad b_2t^2 + b_1t + (b_0 - y) = 0,$$

the resultant of these two polynomials is then defined by the determinant of Sylvester's matrix, namely

$$S = \begin{bmatrix} a_2 & a_1 & a_0 - x & 0 \\ 0 & a_2 & a_1 & a_0 - x \\ b_2 & b_1 & b_0 - y & 0 \\ 0 & b_2 & b_1 & b_0 - y \end{bmatrix}$$

$$\begin{aligned} \Rightarrow |S| = {} & b_2^2x^2 - 2a_2b_2xy + a_2^2y^2 \\ & + \left(2a_2b_2b_0 + a_1b_1b_2 - a_2b_1^2 - 2a_0b_2^2\right)x \\ & + \left(a_2b_1a_1 + 2a_2b_2a_0 - 2a_2^2b_0 - a_1^2b_2\right)y \\ & + (a_2b_0 - a_0b_2)^2 + (a_1b_2 - a_2b_1) \\ & \times (a_1b_0 - a_0b_1) = 0 \end{aligned}$$

an implicit algebraic curve that is equivalent to the parametric curve. In general, if x and y are polynomials of degree p, the corresponding algebraic curve also will have degree p.

To further illustrate this procedure, consider the parametric curve defined by the equations:

$$\begin{aligned} x &= 0.4 + 0.5\cos t - 2\sin 2t \\ y &= 0.6 + 0.2\sin t - 2\sin 2t \end{aligned}$$

By substituting the following equivalent relations for $\cos kt$ and $\sin kt$

$$\cos kt = \frac{e^{ikt} + e^{-ikt}}{2}, \quad \sin kt = \frac{e^{ikt} - e^{-ikt}}{2}$$

and then substituting $z = e^{it}$, the following complex form of these equations is obtained,

$$\begin{aligned} x(z) &= 0.4 + 0.25z + 0.25z^{-1} + iz^2 - iz^{-2} \\ y(z) &= 0.6 - 0.1iz + 0.1iz^{-1} + 0.35z^2 + 0.35z^{-2} \end{aligned}$$

In order to make all z powers positive, both equations are multiplied by z^2, so that

$$\begin{aligned} 0 &= iz^4 + 0.25z^3 + (0.4 - x)z^2 + 0.25z - i \\ 0 &= 0.35z^4 - 0.1iz^3 + (0.6 - y)z^2 + 0.1iz + 0.35 \end{aligned}$$

Using elimination theory, the determinant of Sylvester's matrix is then defined by

$$S = \begin{bmatrix} -i & 0.25 & 0.4 - x & 0.25 & i & 0 & 0 & 0 \\ 0 & -i & 0.25 & 0.4 - x & 0.25 & i & 0 & 0 \\ 0 & 0 & -i & 0.25 & 0.4 - x & 0.25 & i & 0 \\ 0 & 0 & 0 & -i & 0.25 & 0.4 - x & 0.25 & i \\ 0.35 & 0.1i & 0.6 - y & -0.1i & 0.35 & 0 & 0 & 0 \\ 0 & 0.35 & 0.1i & 0.6 - y & -0.1i & 0.35 & 0 & 0 \\ 0 & 0 & 0.35 & 0.1i & 0.6 - y & -0.1i & 0.35 & 0 \\ 0 & 0 & 0 & 0.35 & 0.1i & 0.6 - y & -0.1i & 0.35 \end{bmatrix},$$

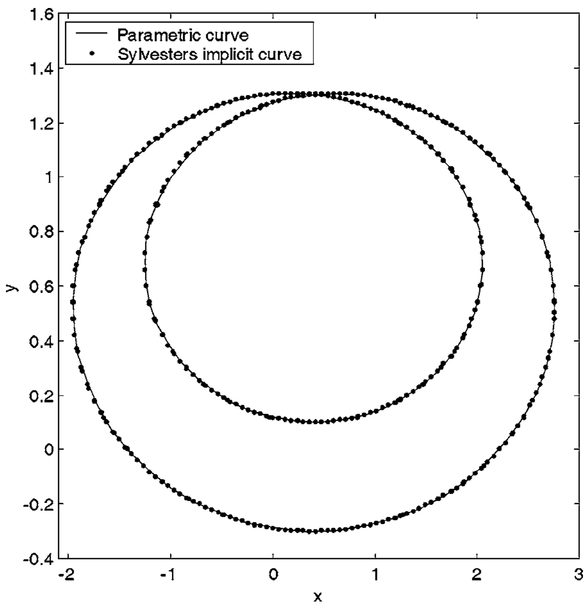

Figure 1. Parametric curve (2 harmonic EFD) and 2nd degree implicit curve (dots) obtained by Sylvester's elimination method are superimposed.

which implies the (monic) quartic implicit polynomial curve

$$\begin{aligned} f_4(x, y) = x^4 &+ 16.327x^2y^2 + 66.639y^4 - 1.6x^3 \\ &- 19.657x^2y - 13.061xy^2 - 160.35y^3 \\ &- 1.332x^2 + 15.726xy + 79.039y^2 \\ &+ 1.578x + 24.698y - 3.736 = 0 \end{aligned}$$

that is shown in Fig. 1.

3. Our Approach: Implicitization by Matrix Annihilation

Consider an n-harmonic elliptic Fourier descriptor representation of any 2-D curve, namely[1]

$$\begin{aligned} x(t) &= a_o + \sum_{k=1}^{n} (a_k \cos kt + b_k \sin kt) \\ y(t) &= c_o + \sum_{k=1}^{n} (c_k \cos kt + d_k \sin kt) \end{aligned} \tag{3.1}$$

where (a_0, c_0) is the center of the curve and $a_k, b_k, c_k, d_k, k = 1, \ldots, n$ are elliptic Fourier coefficients of the curve up to n Fourier harmonics. We next substitute the relations

$$\cos kt = \frac{e^{ikt} + e^{-ikt}}{2}, \quad \sin kt = \frac{e^{ikt} - e^{-ikt}}{2} \tag{3.2}$$

for $\cos kt$ and $\sin kt$ in order to obtain a complex-exponential form of the elliptical Fourier descriptors, namely

$$\begin{aligned} x(t) &= A_0 + \sum_{k=1}^{n} A_k e^{ikt} + B_k e^{-ikt} \\ y(t) &= C_0 + \sum_{k=1}^{n} C_k e^{ikt} + D_k e^{-ikt} \end{aligned} \tag{3.3}$$

where

$$A_k = \frac{(a_k - ib_k)}{2} \quad B_k = \frac{(a_k + ib_k)}{2}$$

and

$$C_k = \frac{(c_k - id_k)}{2} \quad D_k = \frac{(c_k + id_k)}{2}$$

for $k = 1, \ldots, n$. Here $A_0 = a_0$ and $C_0 = c_0$. These equations can be expressed in a more compact form by substituting z for e^{it},

$$\begin{aligned} x(z) &= A_o + \sum_{k=1}^{n} (A_k z^k + B_k z^{-k}) \equiv \sum_{k=-n}^{n} g[k] z^k \\ y(z) &= C_o + \sum_{k=1}^{n} (C_k z^k + D_k z^{-k}) \equiv \sum_{k=-n}^{n} h[k] z^k \end{aligned} \tag{3.4}$$

where

$$g[k] = \begin{cases} A_k & \text{if } k > 0 \\ A_0 & \text{if } k = 0 \\ B_k & \text{if } k < 0 \end{cases} \quad h[k] = \begin{cases} C_k & \text{if } k > 0 \\ C_0 & \text{if } k = 0 \\ D_k & \text{if } k < 0 \end{cases}$$

Alternatively, the g and h sequences can be written as vectors, namely

$$\begin{aligned} g &= [B_n \ \cdots \ B_1 \ A_0 \ A_1 \ \cdots \ A_n] \\ h &= [D_n \ \cdots \ D_1 \ C_0 \ C_1 \ \cdots \ C_n] \end{aligned}$$

Equation (3.4) can then be re-written as

$$x(z) = g \cdot \vec{z} \quad \text{and} \quad y(z) = h \cdot \vec{z}$$

where

$$\vec{z}^T = [z^{-n} \ \cdots \ z^{-1} \ 1 \ z \ \cdots \ z^n].$$

To explicitly determine representations of the monomials, $x^p y^q$, given $x(z) = g \cdot \bar{z}$ and $y(z) = h \cdot \bar{z}$, we next utilize the time convolution property of the well-known z-transform, which states that if

$$g[k] \Leftrightarrow x(z) \quad \text{and} \quad h[k] \Leftrightarrow y(z)$$

then

$$g[k] * h[k] \Leftrightarrow x(z)y(z)$$

Note that convolution in discrete-time domain corresponds to multiplication in the z-domain. For example,

$$\begin{aligned} x^2 &= x(z)x(z) = Z\{g[k] * g[k]\}, \\ xy &= x(z)y(z) = Z\{g[k] * h[k]\}, \\ y^2 &= y(z)y(z) = Z\{h[k] * h[k]\} \end{aligned}$$

The monomials $x^p y^q$ for different p and q values can be found similarly. We can therefore write

$$\underbrace{\begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \\ x^3 \\ x^2y \\ xy^2 \\ y^3 \\ x^4 \\ x^3y \\ \vdots \\ xy^{d-1} \\ y^d \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} 0\cdots0 \quad 1 \quad 0\cdots0 \\ g \\ h \\ g*g \\ g*h \\ h*h \\ g*g*g \\ g*g*h \\ g*h*h \\ h*h*h \\ g*g*g*g \\ g*g*g*h \\ \vdots \\ g*\underbrace{h*\cdots*h}_{d-1} \\ \underbrace{h*h*\cdots*h}_{d} \end{bmatrix}}_{\text{P: Convolution Matrix}} \underbrace{\begin{bmatrix} z^{-nd} \\ \vdots \\ z^{-1} \\ 1 \\ z^1 \\ \vdots \\ z^{nd} \end{bmatrix}}_{\bar{z}},$$

or simply

$$\Gamma = P\bar{z},$$

for some complex $(d+1)(d+2)/2 \times (2dn+1)$ matrix P. We next re-write P as

$$P = \hat{P} \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ i & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & i & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & i & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & i \end{bmatrix}}_{C}$$

for some unique, real $(d+1)(d+2)/2 \times (2d^2+1)$ matrix $\hat{P}$.

We then determine the "largest" $(d+1)(d+2)/2 - 1 = d(d+3)/2$ columns of $\hat{P}$ via an orthogonal-triangular decomposition defined by $QR = \hat{P}E$, where Q is a unitary matrix, R is an upper triangular matrix whose diagonal elements are of decreasing absolute value, and E is a permutation matrix which orders the columns of $\hat{P}E$ in correspondence with those of QR. The unique unit vector ν that annihilates the first $d(d+3)/2$columns of $\hat{P}E$, which we will define as $\tilde{P}$, then yields an appropriate (non-monic) implicit polynomial function as the product $\nu\Gamma = f_d(x, y) = 0$. The Matlab routines "*qr*" and "null" are used to perform the required computations. Implicitization of truncated Fourier descriptors with n harmonics implies an algebraic equation of degree $d = 2n$.

To illustrate our matrix annihilation procedure, consider the earlier parametric curve defined by

$$\begin{aligned} x &= 0.4 + 0.5\cos t - 2\sin 2t \\ y &= 0.6 + 0.2\sin t + 0.7\cos 2t \end{aligned}$$

whose Fourier coefficients are given by

$$\begin{gathered} a_0 = 0.4, \quad c_0 = 0.6 \\ [\,a_1 \quad a_2\,] = [\,0.5 \quad 0\,], \quad [\,b_1 \quad b_2\,] = [\,0 \quad -2\,] \\ [\,c_1 \quad c_2\,] = [\,0 \quad 0.7\,], \quad [\,d_1 \quad d_2\,] = [\,0.2 \quad 0\,] \end{gathered}$$

The complex Fourier coefficients are then determined to be

$$
\begin{aligned}
A_0 = 0.4, \quad C_0 &= 0.6 \\
[\,A_1 \quad A_2\,] &= [\,0.25 \quad i\,], \\
[\,B_1 \quad B_2\,] &= [\,0.25 \quad -i\,] \\
[\,C_1 \quad C_2\,] &= [\,-0.1i \quad 0.35\,], \\
[\,D_1 \quad D_2\,] &= [\,0.1i \quad 0.35\,]
\end{aligned}
$$

The g and h sequences are then defined as

$$
\begin{aligned}
g &= [-i \quad 0.25 \quad 0.4 \quad 0.25 \quad i] \\
h &= [0.35 \quad 0.1i \quad 0.6 \quad -0.1i \quad 0.35]
\end{aligned}
$$

which subsequently imply

$$
\begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \\ \vdots \\ xy^3 \\ y^4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0\cdots0 \quad 1 \quad 0\cdots0 \\ g \\ h \\ g*g \\ g*h \\ h*h \\ \vdots \\ g*h*h*h \\ h*h*h*h \end{bmatrix}}_{P} \begin{bmatrix} z^{-8} \\ \vdots \\ z^{-1} \\ 1 \\ z \\ \vdots \\ z^{8} \end{bmatrix}
$$

The (15×14) $\tilde{P}$ matrix is then determined as outlined above and explicitly given by

1	0	0	0	0	0	0	0	0	0	0	0	0	0
0.4	1	0	0	0	0	0	0	0	0	0	0	0	0
0.6	0	0	0	0.35	0	0	0	0	−0.1	0	0	0	0
2.285	0.8	−1	0	0.0625	0	0	0	0	0.5	0	0	−0.5	0
0.24	0.575	0	0.35	0.14	0	0	0.1875	0	−0.04	0	0	0	0
0.625	0	0.1225	0	0.41	0	0	0	0	−0.05	0	0	0.07	0
2.614	3.855	−1.2	0.1875	0.075	0	−0.75	−0.7344	0	0.6	−1	0	−0.6	0
1.3147	0.46	−0.5281	0.28	0.4872	0.275	0	0.15	−0.35	0.0777	0	0	−0.3687	0
0.25	0.4725	0.049	0.3925	0.164	0	0.1006	0.1831	0	−0.02	0.1225	0	0.028	0
0.672	0	0.21	0	0.5096	−0.0367	0	0	0.0429	−0.0585	0	0	0.0883	0
9.589	5.656	−5.7061	0.3	0.4506	−0.9375	−1.2	−1.175	−0.375	3.605	−1.6	−1	−3.6675	0
1.5009	2.1385	−0.6337	1.1102	0.5399	0.33	−0.5883	0.247	−0.42	0.1061	−0.4594	0	−0.4425	−0.3625
1.1144	0.378	−0.2595	0.314	0.5386	0.3119	0.0805	0.1465	−0.3673	0.0658	0.096	0.1312	−0.2432	0
0.2688	0.4253	0.084	0.4355	0.2039	−0.0147	0.1515	0.2017	0.0171	−0.0234	0.2008	0	0.0353	0.0475
0.7716	0	0.3142	0	0.6174	−0.0696	0	0	0.0955	−0.0617	0	−0.0171	0.1162	0

and the vector that annihilates this matrix is found to be

$$
\begin{aligned}
\nu = [&-0.0191 \quad 0.0081 \quad 0.1265 \quad -0.0068 \quad 0.0806 \\
&0.4049 \quad -0.0082 \quad -0.1007 \quad -0.0669 \\
&-0.8215 \quad 0.0051 \quad 0 \quad 0.0836 \quad 0 \quad 0.3414]
\end{aligned}
$$

Reordering the terms in lexicographic order and dividing by leading coefficient, we obtain the following monic implicit polynomial curve

$$
\begin{aligned}
f_4(x, y) = x^4 &+ 16.327x^2y^2 + 66.639y^4 - 1.6x^3 \\
&- 19.657x^2y - 13.061xy^2 - 160.35y^3 \\
&- 1.332x^2 + 15.726xy + 79.039y^2 \\
&+ 1.578x + 24.698y - 3.736 = 0
\end{aligned}
$$

that is shown in Fig. 2. The annihilated matrix defined by our method is 15×14 and it takes 0.06 seconds to implicitize using Matlab. The Sylvester's matrix is 8×8, but it takes 7.25 seconds to implicitize.

4. Some Additional Examples

In this section, we present the implicitization of more complex curves. To demonstrate the performance of our method when dealing with some possible problems, we employed a number of different curves. Given the ordered sequence of points on each curve, we first computed the parametric Fourier coefficients of the

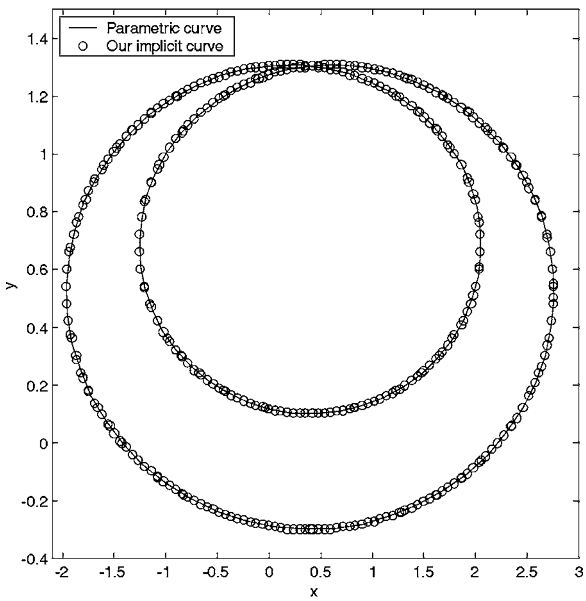

Figure 2. Parametric curve (2 harmonic EFD) and 2nd degree implicit curve (circles) obtained by matrix annihilation method are superimposed.

curve. We then implicitized the curves both by our matrix annihilation method and Sylvester's elimination method. Both implicitization methods were implemented in Matlab.

For Sylvester's elimination method, we used the Symbolic Toolbox of Matlab. Assuming that n harmonic Fourier coefficients are used to represent the curve, our method involves computing the annihilating vector of $\tilde{P}$, whereas implicitization by Sylvester's elimination method involves computing the determinant of a $4n \times 4n$ symbolic matrix. As the number of harmonics used to represent curves increase, the computation time increases for both methods. But because Sylvester's elimination method involves symbolic computation, the computation time increases faster than our method, as we show in following examples.

When the number of harmonics used to represent a curve is greater than 4 (degree of target implicit polynomial greater than 8), we found that implicitization by Sylvester's elimination method is impossible using Matlab because of the excessive size of the symbolic matrices. Thus, in the following examples, implicitization by Sylvester's elimination method is demonstrated only for a curve defined by 3 harmonic Fourier coefficients.

4.1. *Example 1: 3 Harmonic EFD Curve Converted to 6th Degree IP Curve*

The parametric curve in Fig. 3 is represented by following Fourier coefficients:

$$a_0 = -0.0906, \quad c_0 = -0.1194$$

$$[a_1 \quad a_2 \quad a_3] = [-0.1827 \quad -0.1902 \quad -0.1302],$$

$$[b_1 \quad b_2 \quad b_3] = [-0.3267 \quad -0.1753 \quad -0.1568]$$

$$[c_1 \quad c_2 \quad c_3] = [-0.3383 \quad -0.153 \quad -0.0971],$$

$$[d_1 \quad d_2 \quad d_3] = [0.3385 \quad 0.0992 \quad -0.1589]$$

The 6th degree IP obtained by both our annihilation method and Sylvester's elimination method is same, namely

$$\begin{aligned} f(x, y) &= x^6 - 6.4992x^5y + 17.6741x^4y^2 - 25.7403x^3y^3 \\ &\quad + 21.1742x^2y^4 - 9.3283xy^5 + 1.7195y^6 \\ &\quad - 2.5056x^5 + 10.5328x^4y - 16.0141x^3y^2 \\ &\quad + 10.0422x^2y^3 - 1.6236xy^4 - 0.4356y^5 \\ &\quad + 0.1532x^4 + 3.6857x^3y - 7.8187x^2y^2 \\ &\quad + 4.8756xy^3 - 0.7753y^4 + 2.1925x^3 \\ &\quad - 2.9669x^2y + 0.8362xy^2 + 0.0254y^3 \\ &\quad + 0.6077x^2 - 0.5863xy + 0.0803y^2 \\ &\quad - 0.0041x + 0.0599y - 0.0096 = 0 \end{aligned}$$

as shown in Fig. 3. The annihilated matrix $\tilde{P}$ defined by our method is 28×27 and it takes 0.06 seconds to implicitize. The Sylvester's matrix is 12×12, but it takes 8.46 seconds to implicitize.

4.2. *Example 2: 6 Harmonic EFD Curve Converted to 12th Degree IP Curve*

The parametric curve in Fig. 4 is represented by following Fourier coefficients:

$$a_0 = -0.057, \quad c_0 = -0.0157$$

$$[a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6] = [-0.5224 \ -0.0574 \ 0.0830 \ 0.1513 \ 0.0262 \ 0.1703]$$

$$[b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6] = [-0.09 \ 0.089 \ 0.0258 \ -0.0154 \ 0.0251 \ -0.0113]$$

$$[c_1 \ c_2 \ c_3 \ c_4 \ c_5 \ c_6] = [-0.1396 \ 0.0652 \ 0.0682 \ 0.0843 \ -0.0832 \ 0.0176]$$

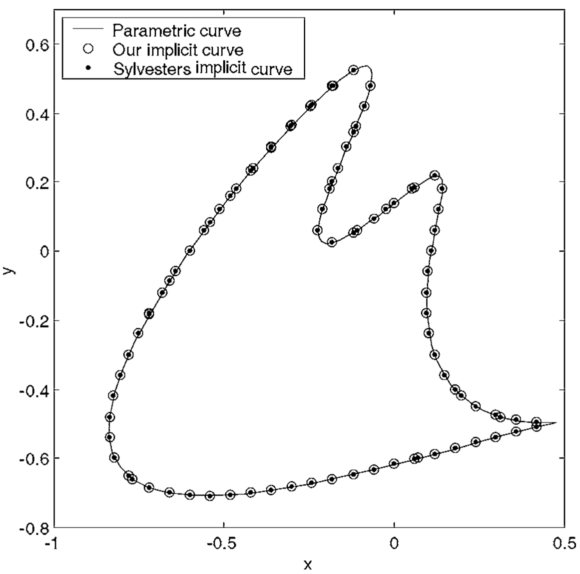

Figure 3. Parametric curve (3 harmonic EFD), 6th degree implicit curve (circles) obtained by matrix annihilation method and, implicit curve (dots) obtained by Sylvester's elimination method are superimposed.

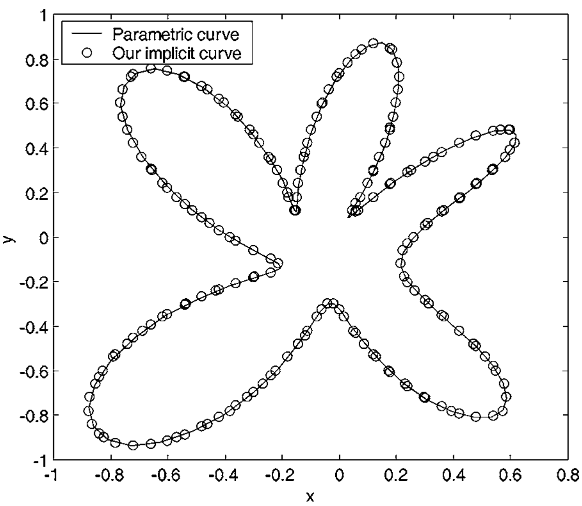

Figure 4. Parametric curve (6 harmonic EFD) and 12th degree implicit curve (circles) obtained by matrix annihilation method are superimposed.

$$[d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6] \\ = [0.6211 \ 0.0893 \ 0.0938 \ 0.1858 \ -0.0864 \ -0.1864]$$

The 12th degree implicit polynomial[2] obtained by our annihilation method is shown in Fig. 4. The annihilated matrix $\tilde{P}$ defined by our method is 91×90 and it takes 1.21 seconds to implicitize. The Sylvester's matrix is 24×24 and, once again, it cannot be implicitized using Matlab or Maple.

5. Implicit Shape Representation in Vision Applications

In this section, we present some results relevant to computer vision. In particular, consider the carpal image of two bones depicted in Fig. 5(a). Using EFD fitting (Lestrel, 1997) followed by our matrix annihilation algorithm, a 10th degree IP curve is determined that describes the boundary of the first bone and an 8th degree IP curve is determined that describes the boundary of the second bone depicted in Fig. 5(b) and (c) respectively. Analogous results are shown in Fig. 6. Similarly, implicit representation of the cell in Fig. 7(a) is shown in Fig. 7(b).

Once an implicit polynomial model that defines the shape of an object has been determined, it can be employed in a variety of ways. In particular, algebraic invariants can easily be computed from the IP equations and subsequently used for object recognition (Wolovich and Unel, 1998; Unel and Wolovich, 1998, 1999, 2000; Subrahmonia et al., 1996).

The classical approach in curvature computation is to use curve fitting or structural models in a local neighborhood of the curve. However, this procedure typically performs poorly in the vicinity of singularities (Monga and Benayoun, 1995). The curvature of an IP-defined object can be directly defined mathematically using derivatives of the IP equations, as shown in Rutter (2000). Such curvature computations can be more direct and accurate than those obtained using the classical approach, especially in the vicinity of singularities. As in the case of invariants, curvature comparisons can be used to define and/or compare similar free-form objects.

Finally, we have recently determined a new and highly accurate method of determining the perpendicular distance from a point in either 2D or 3D space to a curve or surface defined by an implicit polynomial equation. This method has been proven to be very useful in metrology applications (Wolovich et al., 2002). Since the underlying algorithm also can be used to determine the distances between dots and the solid IP curves in Figs. 5–7, it too should prove to be very useful in a variety of model-based vision applications. We are currently investigating all of these methods based on the implicit polynomial

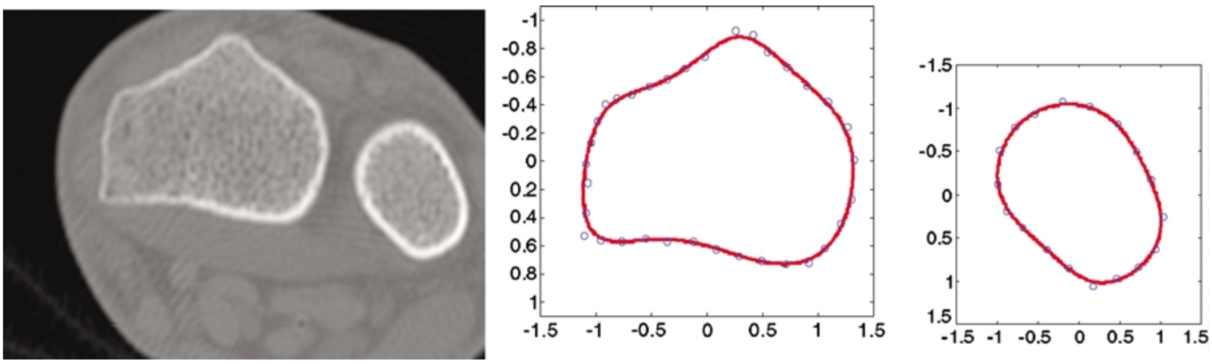

Figure 5. (a) Carpal3d_01_01 image. (b) Contour points of the first bone (dots) and 10th degree IP curve obtained by our matrix annihilation (solid). (c) Contour points of the second bone (dots) and 8th degree IP curve (solid).

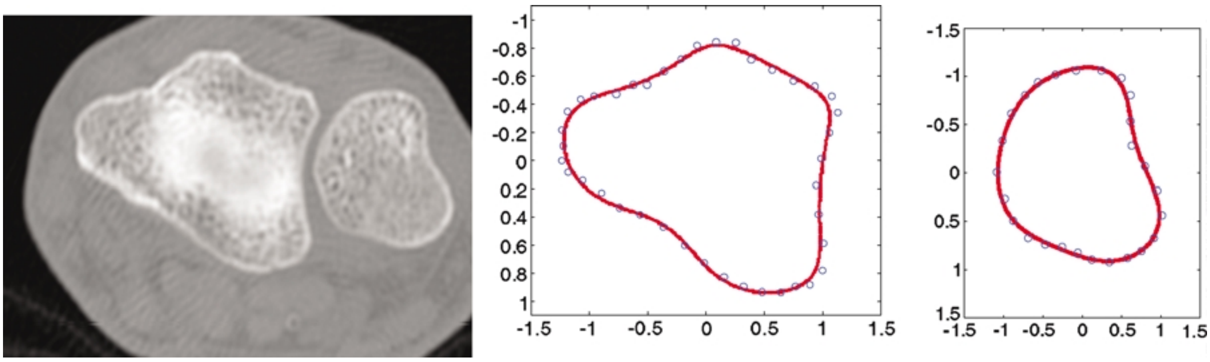

Figure 6. (a) Carpal3d_01_11 image. (b) Contour points of the first bone (dots) and 10th degree IP curve obtained by our matrix annihilation (solid). (c) Contour points of the second bone (dots) and 8th degree IP curve (solid).

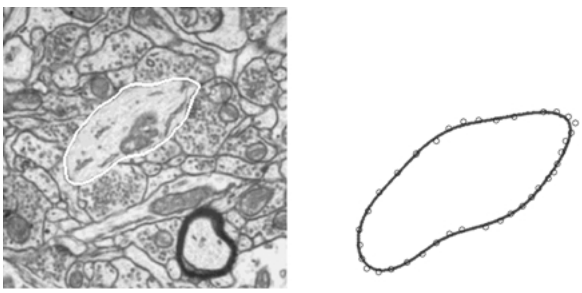

Figure 7. (a) Cell image and (b) contour points of the cell (dots) and 10th degree IP (5 harmonic) (solid).

representations that we obtain using our matrix annihilation procedure.

6. Conclusions

Implicit polynomial representations are very useful for modeling given point data sets, and numerous papers have been written which illustrate their importance in image understanding and object recognition. A variety of methods have been devised for directly fitting given data sets to implicit polynomial curves and, although such methods are continually being refined and improved, alternative implicitization procedures are equally important and useful.

In this report, we have demonstrated a new matrix annihilation method for efficiently converting certain types of parametrically defined curves to algebraic ones. As we have explicitly illustrated, our matrix annihilation method works very well and efficiently in many higher order cases where symbolic-based methods fail, although stability problems and outliers can occur when the number of EFD harmonics used to define an object exceeds six. We believe that alternative, numerical matrix computations may well solve this problem, and we are investigating such a possibility.

As noted earlier, the application of numerical invariants, curvature computations and perpendicular distance approximations, all derived from accurate and robust IP models obtained using matrix annihilation method, can offer significant benefits in many different vision-based applications.

Notes

1. The coefficients of coskt and sinkt in (3.1) can be uniquely determined from an ordered sequence of points which describe the boundary of a curve using the relations given in Chapter 2 of Lestrel (1997). We will assume here that this operation has been performed and, therefore, that the Fourier coefficients are known.
2. The coefficients of the implicit polynomial equation which define this and the remaining curves in this paper can be obtained from the authors.

References

Bajaj, C., Garrity, T., and Warren, J. 1988. On the applications of multi-equational resultants. Technical report CSD-TR-826, Department of Computer Science, Purdue University.

Bloomenthal, J. 1997. *Introduction to Implicit Surfaces.* Morgan Kaufmann Publishers, Inc.

Buchberger, B. 1985. Groebner bases: An algorithmic method in polynomial ideal theory. In *Multidimensional Systems Theory*, N.K. Bose (Ed.), D. Reidel Publishing Company, pp. 184–232.

Buchberger, B. 1989. Applications of groebner bases in non-linear computational geometry. In *Geometric Reasoning*, D. Kapur and J. Mundy (Eds.), MIT Press, pp. 415–447.

Calabi, E., Olver, P.J., Shakiban, C., Tannenbaum, A., and Haker, S. 1998. Differential and numerically invariant signature curves applied to object recognition. *IJCV*, 26(2):107–135.

Canny, J.F. 1990. Generalized characteristic polynomials. *Journal of Symbolic Computation*, 9:241–250.

Chionh, E.W. 1990. Base points, resultants, and the implicit representation of rational surfaces. Ph.D. Thesis, Department of Computer Sicence, University of Waterloo, Canada.

Chionh, E.W. and Goldman, R.N. 1992. Using multivariate resultants to find the implicit equation of a rational surface. *The Visual Computer*, 8(3):171–180.

Dixon, A.L. 1908. The eliminant of three quantics in two independent variables. *Proc. London Mathematical Society*, *Series 2*, 7:49–69.

Granlund, G.H. 1972. Fourier preprocessing for hand print character recognition. *IEEE Trans. Computers*, 21:195–201.

Hobby, J.D. 1991. Numerically stable implicitization of cubic curves. *ACM Transactions on Graphics*, 10(3).

Hoffmann, C. 1989. *Geometric and Solid Modeling: An Introduction.* Morgan Kaufmann Publishers Inc.

Hoffmann, C. 1993. Implicit curves and surfaces in CAGD. *IEEE Computer Graphics and Applications*, 13(1):79–88.

Hong, H. 1995. Implicitization of curves parametrized by generalized trigonometric polynomials. In *Proceedings of Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC-11)*, pp. 285–296.

Huang, Z.H. and Cohen, F.S. 1996. Affine invariant B-spline moments for curve matching. *Image Processing*, 5(10):1473–1480.

Keren, D., Cooper, D., and Subrahmonia, J. 1994. Describing complicated objects by implicit polynomials. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:38–53.

Kuhl, F.P. and Giardina, C.R. 1982. Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processing*, 18:236–258.

Lei, Z., Blane, M.M., and Cooper, D.B. 1996. 3L fitting of higher degree implicit polynomials. In *Proceedings of Third IEEE Workshop on Applications of Computer Vision*, Florida, pp. 148–153.

Lestrel, P.E. 1997. *Fourier Descriptors and Their Applications in Biology.* Cambrigde Univ. Press.

Lin, C.S. and Hwang, C.L. 1987. New forms of shape invariants from elliptic Fourier descriptors. *Pattern Recognition*, 20(5):535–545.

Ma, S. 1993. De. Conics-based stereo, motion estimation and pose determination. *IJCV*, 10 (1).

Macaulay, F.S. 1916. *The Algebraic Theory of Modular Systems.* Cambridge University Press.

Manocha, D. and Canny, J.F. 1992a. The implicit representation of rational parametric surfaces. *Journal of Symbolic Computation*, 13:485–510.

Manocha, D. and Canny, J.F. 1992b. Algorithms for implicitizing rational parametric surfaces. *Journal of Computer-Aided Geometric Design*, 9:25–50.

Monga, O. and Benayoun, S. 1995. Using partial derivatives of 3D images to extract typical surface features. *Computer Vision and Image Understanding*, 61(2):171–189.

Mundy, J.L. and Zisserman, A. (Eds.) 1992. *Geometric Invariance in Computer Vision.* MIT Press.

Rutter, J.W. 2000. *Geometry of Curves* Chapman & Hall/CRC.

Salmon, G. 1915. *A Treatise on the Analytic Geometry of Three Dimensions.* Longmans, Green anc Co., London, 5th ed., p. 264.

Sederberg, T.W. 1983. Implicit and parametric curves and surfaces for computer aided geometric design. Ph.D. Thesis, Dept. Mechanical Engineering, Purdue University.

Sederberg, T.W. and Anderson, D.C. 1984. Implicit representation of parametric curves and surfaces. *Comput. Vision & Graph. Image Process.*, 28(1):72–84.

Sederberg, T. and Chen, F. 1995. Implicitization using moving curves and surfaces. In *Proceedings of SIGGRAPH*, pp. 301–308.

Sederberg, T.W. and Goldman, R.N. 1986. Algebraic geometry for computer-aided geometric design. *IEEE Comput. Graph. Appl.*, 6(6):52–59.

Solina, F. and Bajcsy, R. 1990. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 12(2):131–147.

Subrahmonia, J., Cooper, D.B., and Keren, D. 1996. Practical reliable bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):505–519.

Tasdizen, T., Tarel, J.P., and Cooper, D.B. 2000. Improving the stability of algebraic curves for applications. *IEEE Transactions in Image Processing*, 9(3):405–416.

Taubin, G. 1991. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1115–1138.

Taubin, G., Cukierman, F., Sullivan, S., Ponce, J., and Kriegman, D.J. 1994. Parameterized families of polynomials for bounded algebraic curve and surface fitting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(3):287–303.

Unel, M. and Wolovich, W.A. 1998. Pose estimation and object identification using complex algebraic representations. *Pattern Analysis and Applications Journal*, 1 (3).

Unel, M. and Wolovich, W.A. 1999. A new representation for quartic curves and complete sets of geometric invariants. *Int. Jour. of Pattern Recognition and Artificial Intelligence*, 13(8).

Unel, M. and Wolovich, W.A. 2000. On the construction of complete sets of geometric invariants for algebraic curves. *Advances in Applied Mathematics*, 24:65–87.

Unsalan, C. and Ercil, A. 2001. Conversions between parametric and implicit forms using polar/spherical coordinate representations. *Computer Vision and Image Understanding*, 81:1–25.

Wallace, T.P. and Mitchell, O.R. 1980. Analysis of three-dimensional movement using fourier descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):583–588.

Wolovich, W., Albakri, H., and Yalcin, H. 2002. The precise modeling and measurement of free-form surfaces. *Journal of Manufacturing Science and Engineering*.

Wolovich, W.A. and Unel, M. 1998. The determination of implicit polynomial canonical curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1080–1090.

Wu, M.F. and Sheu, H.T. 1998. Representation of 3D surfaces by two-variable Fourier descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(8):583–588.

Zahn, C.T. and Roskies, R.Z. 1972. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, 21(3):269–281.