New Algebraic Invariants of Implicit Polynomials for 3D Object Recognition[1]

B. Borhan and A. Erçil

Sabanci University, Faculty of Engineering and Natural Sciences, Istanbul, Turkey
e-mail: aytulercil@sabanciuniv.edu

Abstract—In this paper, we present a method for deriving the rotation invariants of second- and fourth-degree implicit polynomials, and we build a system for 3D object recognition using the derived invariants. Our results show that invariants derived in this paper are stable and the success of the recognition is high when the polynomial fit is successful.

1. INTRODUCTION

Object recognition is a major task for many computer-controlled systems. It is used in many industrial applications, such as guiding of robots, sorting products, and in inspection applications. 3D object recognition has always been a challenge in machine vision systems because of the complexity of data and the calculations. However, with the development of more and more powerful and faster systems, today 3D object recognition is possible in real time.

Implicit 2D curves and 3D surfaces are believed to be among the most powerful shape representations presently known. With this approach, objects in 2D images are described by their silhouettes and then represented by 2D implicit polynomial (IP) curves, while objects in 3D data are represented by the IP surfaces.

Invariants are properties of geometric configurations that remain unchanged under an appropriate class of transformations and hence are good descriptors for recognition.

The goal of this research is to derive the invariants of IPs and analyze their usage in object recognition systems.

Through the proposed concepts and algorithms, we will argue that invariants of implicit polynomials provide a fast and stable model for 3D recognition problems stemming from industrial inspection.

The application we are interested in is recognizing a set of objects on a conveyor belt. Thus, the transformations under consideration are rotation around one specific axis, y, and translation, although the technique used can be generalized to other transformations like affine, perspective, etc. The translation problem has been overcome by finding the center of the objects from the point data set and aligning the object centers with the world origin; hence, the invariance sought in this paper is for rotation around one axis.

Section 2 summarizes the implicit polynomial model and the 3L fitting [8] method. In Section 3, we give a brief description of the invariant theory and explain the symbolic computation method for finding algebraic invariants of IPs. In Section 4, we present our test results with a discussion on the success of our work.

Finally, in Section 5, we conclude by discussing the advantages and disadvantages of the symbolic computation method and propose some future work for increasing the success of recognition.

2. IMPLICIT POLYNOMIALS

IPs, being among the most effective and leading shape representations for complex free-form object modeling and recognition, have found wide application areas in computer vision. Although the underlying theory, i.e., algebraic geometry, has long been around, IPs could not find effective application areas until the 1980s. Since then, independent research in a number of fields, including computer graphics, geometric modeling, and computer vision, has accumulated valuable insights into various properties of IPs important for solving practical problems.

An implicit polynomial of degree n $f(x, y) = \sum_{i, j \geq 0, i+j \leq n} a_{ij} x^i y^j = 0$ is assumed to represent a shape (object) $S = \{(x_k, y_k)|k = 1, \ldots, M\}$ if every point of the shape S is on the zero set of the implicit polynomial $Z(f) = \{(x, y)|f(x, y)\} = 0$. Robust and consistent IP fits to data sets is the most important requirement for the practicality of IP-related techniques. A variety of iterative and noniterative solution techniques, perturbation techniques, and stopping rules have been proposed to acquire this goal. The general IP fitting problem can be set up as follows. Given a data set $r_0 = \{(x_m, y_m, z_m)|m = 1, \ldots, K\}$, find the nth-degree IP $f_n(x, y, z)$ that minimizes the average squared distance from the data points

[1] This article was submitted by the authors in English.

Received October 25, 2004

(a) (b) (c)

Fig. 1. (a) Teapot object without the spout and lid; (b) level sets for the modified teapot object; (c) fourth-degree fit to the object.

to the zero set $Z(f)$ of the polynomial [13]. There is no closed form expression for the distance from a point to a generic implicit curve or surface, not even for algebraic curves or surfaces, and iterative methods are required to compute it.

Many good algorithms have been presented for solving this problem to get the best fitting polynomial [6, 12]. Taubin [12] developed an approximate distance from a point to a curve or surface defined by implicit equations, changing the problem of fitting curves and surfaces into the minimization of the approximate mean square distance. Then, he showed that the minimization of the approximate mean square distance reduces to a generalized eigenvector computation for certain families of nonsingular curves and surfaces and introduced an efficient procedure to compute an initial estimate for the general case based on these results [12]. IPs, bounded or unbounded, even though represent the data very well, have a very large data set, and this is an important difficulty with them. Keren, Cooper, and Subrahmonia presented a model for fitting polynomials, particularly fourth-degree IPs whose zero set is bounded, stable, and "tight" around the object [6]. Lei and Cooper [9] presented a general framework for using linear programming technology to solve fitting problems, which allowed users to interactively choose significant points such that the fitted curve lies within a user-specified ε using the linearized distance approximation. Finally, the 3L algorithm provides a linear solution to the fitting problem while overcoming many drawbacks of the previous algorithms. The presence of singularities of the polynomial f in the vicinity of the data set, Γ_0, is an important disadvantage of IP fitting. If we define $d(x, y, z)$ as the function which, at (x, y, z), takes on the value of the signed distance from (x, y, z) to Γ_0 and fit the explicit polynomial $f(x, y, z)$ to a portion of the distance transform $d(x, y, z)$ of Γ_0, we get fast, stable, and repeatable IP surface fits [3]. Other than the original data set Γ_0, 3L fitting also uses a pair of synthetically generated data sets Γ_{+c} and Γ_{-c} consisting of points at a distance Γ_c to either side of Γ_0 (Fig. 1); hence, it is called 3L or three-level. Figure 1b shows the three-level sets for the modified teapot object in 3D, and the fourth-degree IP fit is shown in Fig. 1c.

A typical application of IPs in computer vision has been object recognition. For this purpose, in the 3D case, first the 3D data of the object is generated using a 3D image acquisition system such as stereo vision, structured light, or a laser scanner. Then, an IP surface is fit to the data for representation, and a vector of algebraic invariants are computed from the coefficients of the IP to form the invariant space representation of the object. This vector is compared with vectors in a database for solving the matching problem.

3. INVARIANTS AND OBJECT RECOGNITION

The essence of our approach to object recognition is to derive properties of the object geometry which are reliably invariant to the transformation of interest from image intensity data and describe the objects in terms of such invariants, which provide all of the essential information about shape and configuration required to carry out visual tasks.

3.1. Using Symbolic Computation to Find Algebraic Invariants of 3D Objects

In 1994, Keren [5] proposed a symbolic calculation tool to find simple and explicit invariants of polynomials by assuming that the invariants are low-degree homogeneous polynomials in the coefficients. The degree of the homogeneous polynomial is called the rank of the invariant. We have applied the symbolic calculation method proposed in [5] in the calculation of rotation invariants of 3D objects with ranks 2 and 4. We have used the MatCAD and MatLAB packages for symbolic calculations. If we denote a polynomial with three variables as

$$P(x, y, z) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} \sum_{k=0}^{n-j-i} p_{ijk} x^i y^j z^k, \tag{3.1}$$

and x, y, and z are subject to some kind of a transformation $(u, v, w)^t = T(x, y, z)^t$, where T is determined by a certain number of parameters t_{ij}. Then, $P(x, y, z)$ transforms into a polynomial $Q(u, v, w)$, where Q's parameters q_{ij} are functions of p_{ij} and t_{ij}. From now on, we will index coefficients p_{ij} and q_{ij} with a single variable for convenience. So, $P(x, y, z)$ is determined by the coefficients $\{p_i\}_{i=1}^{i=N}$ and $Q(u, v, w)$ is determined by the coefficients $\{q_i\}_{i=1}^{i=N}$, where each q_i is a function of p_i's and t_{ij}'s.

Now, we assume a particular algebraic structure for the invariant I, which is a homogeneous polynomial ψ in the p_i's. From the theory of invariance, the following equation must be true:

$$\begin{aligned} I &= \sum_{o \le i < j < k \le N} \Psi_{ijk} p_i p_j p_k \\ &= \sum_{o \le i < j < k \le N} \Psi_{ijk} q_i q_j q_k. \end{aligned} \tag{3.2}$$

For example, if we consider rotation by an angle θ, each q_i will be a function of p_i and θ. For example, if the degree of the polynomial is 4 ($d = 4$), then the polynomial

$$\begin{aligned} P_h(x, y, z) &= p_{040}y^4 + p_{202}x^2z^2 + p_{400}x^4 + p_{301}x^3z \\ &+ p_{211}x^2yz + p_{310}x^3y + p_{121}xy^2z + p_{113}xyz^2 + p_{130}xy^3 \\ &+ p_{031}y^3z + p_{103}xz^3 + p_{022}y^2z^2 \\ &+ p_{013}yz^3 + p_{220}x^2y^2 + p_{004}z^4 \end{aligned} \tag{3.3}$$

is transformed into the polynomial

$$\begin{aligned} Q_h(u, v, w) &= p_{040}v^4 + q_{202}u^2w^2 + q_{400}u^4 + q_{301}u^3w \\ &+ q_{211}u^2vw + q_{310}u^3v + q_{121}uv^2w + q_{113}uvw^2 \\ &+ q_{130}uv^3 + q_{031}v^3w + q_{103}uw^3 + q_{022}v^2w^2 \\ &+ q_{013}vw^3 + q_{220}u^2v^2 + q_{004}w^4, \end{aligned} \tag{3.4}$$

and the transformed $[u, v, w]$ coordinates will be calculated as follows:

$$\begin{aligned} u &= x\cos\theta - z\sin\theta, \\ v &= y, \\ w &= x\sin\theta + z\cos\theta. \end{aligned} \tag{3.5}$$

If we do the substitutions in Eqs. (3.5) and replace $\sin\theta$ and $\cos\theta$ by their first-order Taylor approximations, θ and 1, and discard all powers of θ higher than 1, we get a linear system of equations between p's and q's.

If we look for invariants Ψ which are second-degree polynomials in p_{ijk}, the solution to the following equation has to hold for every angle θ and every coefficient vector p:

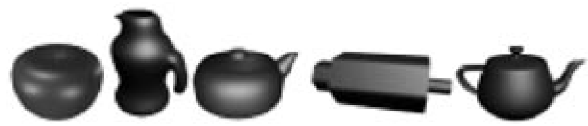

Fig. 2. Some objects from the database.

$$\begin{aligned} \Phi(\{p_i\}) &= \sum_{0 \le j < j < k < N} \Phi_{ijk}p_ip_jp_k \\ &= \sum_{0 \le j < j \le N} \Phi_{ijk}q_iq_jq_k = \Phi(\{q_i\}). \end{aligned} \tag{3.6}$$

For Eq. (3.6) to hold, it is necessary and sufficient for every p,

$$\left(\frac{\partial}{\partial\theta}\Psi[\Phi(\theta, p)]\right), \tag{3.7}$$

where $\Psi[\Phi(\theta, p)]$ is the homogeneous polynomial. The solution to Eq. (3.7) will give us the coefficients of our invariant equation.

A second-degree y-axis rotation invariant of a fourth-degree IP with three variables is given below:

$$\begin{aligned} \Psi_1^4 &= 6.500_{p040p022} + 6.500_{p040p220} + 1.480p_{301}^2 \\ &+ 0.986p_{202}^2 + 2.840p_{121}^2 + 2.065p_{211}^2 + 1.480p_{103}^2 \\ &+ 5.918p_{004}^2 + 2.065p_{112}^2 + 3.150p_{031}^2 + 6.196p_{310}^2 \\ &+ 6.196p_{013}^2 + 5.680p_{220}^2 + 5.918p_{400}^2 \\ &+ 5.680p_{022}^2 + 3.150p_{130}^2. \end{aligned}$$

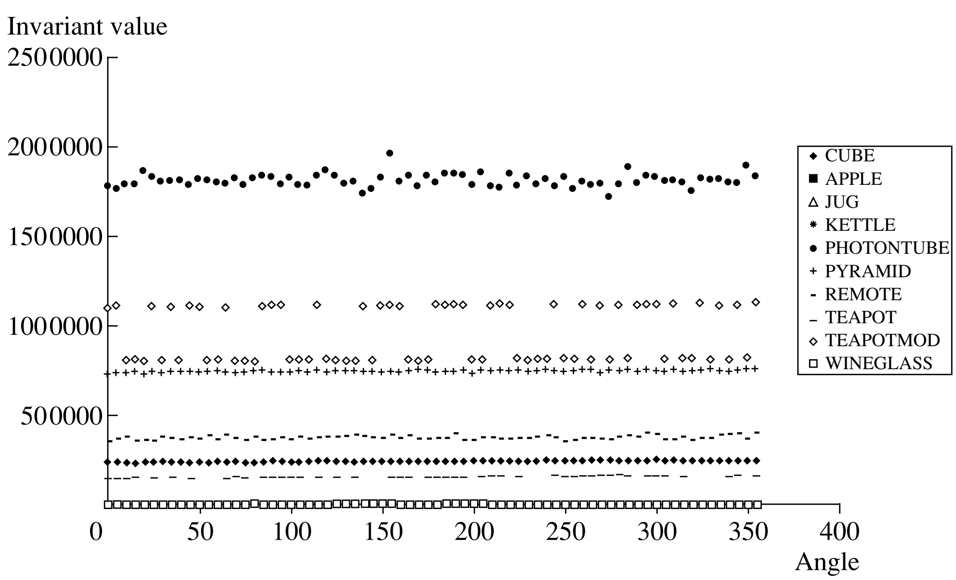

Fig. 3. The distribution of invariant 5 according to the viewing angle for fourth-degree fit.

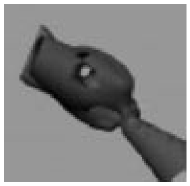

Fig. 4. Fourth-degree 3L fitting for one of the transformed teapot objects.

Other invariants obtained are given in [1].

4. EXPERIMENTAL RESULTS

We have developed a software, Recognition Studio, for testing our invariants and running recognition tests on our test objects using the derived invariants. The software is developed using Microsoft Visual Studio 6.0 and the Visualization ToolKit (VTK), which is an open source freely available software system for 3D computer graphics, image processing, and visualization.

The objects we have used for testing were created by 3D Max Studio 5 3D modeling software. We converted the max files to 3D point sets and stored them in point files to be accessed from our recognition software. Some of the images of the objects in the database are presented in Fig. 2.

For the recognition process, we first teach each object to the database by calculating the invariants from six different views: 0, 60, 120, 180, 240, and 300 degrees.

The success of our work is determined by the standard deviation of invariants of the object for different rotations around the y-axis. This standard deviation is both effected by the polynomial fit and by the calculation of invariants. Figure 3 shows the distribution of one of the invariants for several of the objects in the database.

The table shows the mean and standard deviation of the invariants of two of the objects in the database. As can be seen from Fig. 3 and the table, most of the invariants have low variance and are discriminatory for the objects in the test set.

Seventy-two transformations for each object have been done for the test phase. The recognition rate has been 100% for all the objects except for the teapot and the wineglass objects. Figure 4 shows the fourth-degree 3L fitting for one of the transformed teapot objects. As can be seen from the figure, this fitting is not very successful and is thought to be the reason for the poor performance for the recognition of this object.

The mean and standard deviation of invariants of the kettle and jug objects

	Kettle		Jug	
Invariant	Mean	Standard deviation	Mean	Standard deviation
1	17845.8182	41.8458	183.9181	16.0197
2	16875.5381	39.2842	175.7905	15.3834
3	12532.4808	7.9976	131.0283	11.7559
4	19052.8598	55.1994	193.9410	16.9200
5	17872.1872	40.1814	177.8938	15.7241
6	12867.3472	31.0550	141.1093	12.0820
7	14851.6482	11.9914	143.6977	13.2189
8	22594.3656	59.2775	218.0282	19.3234
9	16475.8983	32.8862	158.5809	14.3021
10	24201.7363	66.7879	250.8086	21.4895
11	17324.6786	41.7908	181.5914	15.6722

5. CONCLUSIONS

In this paper, we described a symbolic computation method for calculating the algebraic invariants of IPs for 3D object recognition. To find the algebraic invariants of IPs for 3D objects using symbolic computation, we first take a general IP of selected degree (P) for the representation of a 3D object. A transformation matrix, for which the invariants are needed, is defined and the transformation is applied to the polynomial to calculate the transformed IP. Then, a particular algebraic structure for the invariant having the coefficients of IPs as variables is assumed. The coefficients of the invariant polynomial are then calculated using the invariant theorem, which is based on the invariance of the rotated polynomial P, with respect to any rotation angle. As the number of variables is higher than the number of equations, symbolic computation is used to calculate the invariant equations. Then, the calculated invariants are used for recognition of objects.

The preliminary experiments in this paper suggest that invariants derived in this paper are stable, and the success of the recognition is high when the polynomial fit is successful.

Future work will include deriving invariants of different-degree polynomials and deriving invariants under different transformation groups. The invariants will also be tested with more real data, including data from some standard 3D databases.

REFERENCES

1. B. Borhan, *Algebraic Invariants of Implicit Polynomials for 3D Object Recognition*, PhD Thesis (Boğazici Univ., 2003).
2. H. Civi, C. Christopher, and A. Ercil, “The Classical Theory of Invariants and Object Recognition Using Algebraic Curves and Surfaces,” J. Math. Imaging Vis. **19** (3), 239–255 (2003).
3. P. Danielsson, E., “Euclidean Distance Mapping,” Comput. Graph. Image Process. **14**, 227–248 (1980).

4. D. Forsyth, A. Z. C. C. A. H., J. L. Mundy, and C. Rothwell, “Invariant Descriptors for 3D Object Recognition and Pose,” IEEE PAMI **13** (10), 971–991 (1991).

5. D. Keren, “Using Symbolic Computation to Find Algebraic Invariants,” IEEE Trans. Pattern Anal. Machine Intell. **16** (11), 1943–1949 (1994).

6. D. Keren, D. C., and J. Subrahmonia, “Describing Complicated Objects by Implicit Polynomials,” IEEE PAMI **16** (1), 191–210 (1994).

7. D. J. Kriegman and J. Ponce, “On Recognizing and Positioning Curved 3D Objects from Image Contours,” IEEE PAMI **12** (12), 1127–1138 (1990).

8. M. M. Blane, Z. Lei, H. Civi, and D. B. Cooper, “The 3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data,” IEEE PAMI **22** (3), 298–313 (2000).

9. Z. Lei and D. B. Cooper, “Linear Programming Fitting of Implicit Polynomials,” IEEE PAMI **20** (2), 212–217 (1998).

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

11. J. L. Mundy and A. Zisserman, “Towards a New Framework for Vision,” *Geometric Invariance in Computer Vision*, Ed. by J. L. Mundy and A. Zisserman (MIT Press, 1992), Vol. 1, pp. 1–39.

12. G. Taubin, “Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations, with Applications to Edge and Range Image Segmentation,” IEEE PAMI **13** (11), 1115–1138 (1991).

13. G. Taubin, S. S. J. P., F. Cukierman, and D. Kriegman, “Parametrized Families of Polynomials for Bounded Algebraic Curve and Surface Fitting,” IEEE PAMI **16** (3), 287–303 (1994).