2D OBJECT RECOGNITION USING IMPLICIT POLYNOMIALS AND ALGEBRAIC INVARIANTS

Uğur Murat ERDEM *Hakan ÇİVİ*[1] *Aytül ERÇİL*

Bogaziçi University, Department of Industrial Engineering
BUPAM Laboratory
80815, Bebek, Istanbul
erdemu@boun.edu.tr civi@lems.brown.edu ercil@boun.edu.tr

ABSTRACT

The problem studied in this paper is the recognition of free-form objects from their 2D visual data in manufacturing environment and the integration of CAD systems with recognition systems. A system based on implicit polynomials and algebraic invariants is developed. Integration of such a visual system with the computer aided design (CAD) phase of the production has been attempted allowing the data exchange in between. To allow this exchange, data produced by CAD system had to be converted to some format in order to be acceptable by the recognition system. Since many CAD packages use parametric form for representing designed objects, the conversion from parametric representation to its implicit correspondent is studied. Even if the conversion proved to be successful, problems between fitted and converted implicit polynomials arose. So this topic is left as a research topic for future studies.

1. INTRODUCTION

In a number of computer vision applications, the goal of the vision system is to locate a specified object in the scene. Such *a priori* knowledge of the object is provided through a *model* of the object. Most existing vision systems rely on models generated in an ad hoc manner and have no explicit relation the CAD/CAM system originally used to design and manufacture these objects. As a result of the different modeling requirements for CAD and vision systems, currently most systems supporting CAD do not provide vision capabilities. Similarly, vision systems incorporate no explicit relation to CAD models. What is desired is a systematic approach for both the generation of representations and recognition strategies based on the CAD models. Such a system provides an integrated automation environment.

There are many techniques available to describe object boundaries, such as B-splines, NURBS (Non-uniform Rational B-Splines), Fourier descriptors, chain codes, polygonal approximation, curvature primal sketch, medial axis transform. Implicit polynomials are among the most effective representations for complex free-form boundry object modeling and recognition with their stability, robustness and invariant characteristics. These features are highly favorable for machine vision applications where distortions (due to lighting) and transformations (such as rotation or scaling) of objects under consideration, are highly expected. Results obtained in laboratory environment show that object recognition systems based on implicit polynomials could be very successful.

2. SYSTEM COMPONENTS

The recognition system developed in BUPAM (Boğazici University Pattern Analysis and Machine Vision Laboratory) and based on implicit polynomials is called ORECIM (Object Recognition Using Implicit Polynomials). Basic components of the system are:

a) Capturing the image data of the object
b) Image processing
c) Implicit polynomial fitting
d) Parametric-implicit conversion
e) Recognition

2.1. Image Capture and Processing

ORECIM uses homogenous background illumination method as it gives high contrast between background and the object, necessary for extracting contour data in the following step. Images are 256 gray-level.

To extract the contour data of the object from the captured image, ORECIM applies thresholding and edge detection methods.

Marr-Hildreth edge detection algorithm is used in order to find edges in the binary image obtained after thresholding.

[1] Currently at Brown University LEMS Laboratory

2.2. Implicit Polynomial Fitting

The contour data prepared by the image processing component of ORECIM is standardized and must be represented by a robust mathematical model, implicit polynomials in ORECIM case. To improve the robustness and uniqueness of the comparison of the unknown data with the model in the data-base, it is a good idea to convert the data to a standard position relative to some its property. One of the approaches, which is also used in ORECIM, is to bring the centroid of the boundary data to point (0,0), the center of the coordinate system.

An implicit polynomial of degree *N* is a polynomial function $f(x,y)=0$ where

$$f(x,y)=\sum_{i,j\geq 0, i+j\leq N} a_{ij}x^i y^i$$

"Implicit polynomial fitting" is the task to find the implicit polynomial coefficients that minimize its distance to the data points. ORECIM uses "Generalized Eigenvalue Fitting" (GEF) algorithm [9] or "3L Fitting" algorithm [1] in order to find such a polynomial. Figure 2 shows a 4th degree GEF for some object.

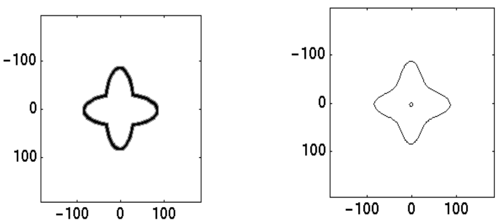

Figure 2: Contour data of the object (left) and 4th degree implicit polynomial fit (right)

Two methods are proposed in order to overcome the difficulties caused by the singularity of the Jacobian matrix used in GEF. Let us define X for a 4th degree polynomial as follows:

$$X=\left(x^4,x^3y,x^2y^2,xy^3,y^4,x^3,x^2y,xy^2,y^3,x^2,xy,y^2,x,y,1\right)^t$$

GEF requires the computation of the Jacobian of X (DX) which always results in a singular one because of the existence of the constant term coefficient.

$$DX=\begin{bmatrix} 0 & x^3 & 2yx^2 & 3y^{2x} & 4y^3 & 0 & x^2 & 2yx & 3y^2 & 0 & x & 2y & 0 & 1 & 0 \\ 4x^3 & 3x^2y & 2xy^2 & y^3 & 0 & 3x^2 & 2xy & y^2 & 0 & 2x & y & 0 & 1 & 0 & 0 \end{bmatrix}^t$$

First method proposes the elimination of the constant term from the polynomial. This will give an X vector without the last constant term, resulting in a Jacobian not necessarily singular. One interesting aspect of this method is that, with 2nd degree implicit polynomials GEF gave 2 perpendicular lines passing from the axes of the data points for objects in our data base. Figure 3 shows such a fit for the same data set in Figure 1.

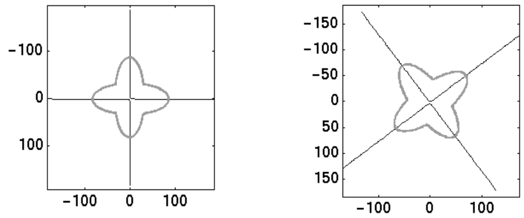

Figure 3: 2nd degree GEF using first approach (left), same fit to the rotated data set (right)

2nd method proposed is the addition of a qxq sparse matrix S with all elements equal to 0 except $S_{q,q}$ equal to 1, to the N matrix used in GEF:

$$N=\frac{1}{n}\sum_{(x_i,y_i)\in Z} DX(x_i,y_i)D^t X(x_i,y_i)$$

$$S_{i,j}=\begin{matrix} 1 & i=j=q \\ 0 & otherwise \end{matrix}$$

where q is the number of coefficients of the implicit polynomial. This again prevents N to be necessarily singular. Figure 4 shows the 4th degree GEF to the pliers data using second approach.

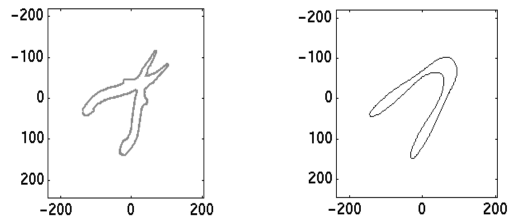

Figure 4: Pliers data (left), 4th degree GEF using 2nd approach

After the fitting procedure the coefficient vector obtained is sent to the recognition step of the system. The shape of the object is now represented by this vector with different size depending on the degree of the implicit polynomial.

2.3. Recognition

In order to make a proper recognition using the implicit polynomial coefficients it is necessary to find and extract robust features from the coefficient vector.

The raw coefficients are not suitable for recognition purposes since any minor perturbation in the data set would result in changes in the coefficient vector also. ORECIM currently utilizes "algebraic invariants" which are a very important characteristics of the implicit polynomials as features. Algebraic invariants are special mappings from coefficient vector (R^q) to R^1. Algebraic invariants of implicit polynomials can be found for Euclidean transformations and affine transformations [2]. The translation invariance of the features is not fully utilized since the data set is currently normalized to the center of the coordinate system.

In the first set of experiments Euclidean invariants of a 4^{th} degree implicit polynomial due to Keren [5] are used:

$$inv_1 = 3a_{13}^2 - 8a_{04}a_{22} + 2a_{13}a_{31} + 3a_{31}^2 - 32a_{40}a_{04} - 8a_{22}a_{40}$$

$$inv_2 = 3a_{04}^2 + 2a_{04}a_{22} + a_{13}a_{31} + 2a_{04}a_{40} + 2a_{22}a_{40} + 3a_{40}^2$$

$$inv_3 = a_{22}^2 - 3a_{13}a_{31} + 12a_{04}a_{40}$$

where a_{ij} is the coefficient of the monomial which is i^{th} degree with respect to x and j^{th} degree with respect to y (i+j=q).

Invariant values found by rotating the pliers data from 0 to 90 degrees with 5 degree increments are given in Table 1.

Rotation	Inv 1 $x10^{-14}$	Inv 2 $x10^{-14}$	Inv 3 $x10^{-15}$
0	1.52	6.90	5.37
5	1.50	6.86	5.02
10	1.56	7.18	4.32
15	1.50	6.92	5.13
20	1.53	6.70	5.50
25	1.54	6.73	5.25
30	1.53	6.90	5.43
35	1.53	6.91	4.58
40	1.55	7.33	3.87
45	1.53	7.03	5.68
50	1.52	7.13	5.27
55	1.53	6.75	5.92
60	1.54	7.05	4.69
65	1.54	7.03	4.37
70	1.57	6.81	5.24
75	1.56	6.98	5.12
80	1.56	6.98	4.66
85	1.52	6.85	5.33
90	1.52	6.89	5.37
μ	1.53	6.94	5.06
σ	3.71e-18	25.18e-17	28.33e-17

Table 1: 3 invariant values for rotated pliers data.

As can be seen from Table 1, invariant values are very close for rotated versions of the GEF. This encourages the use of invariants for recognition purposes.

The criteria used for discriminating between objects in ORECIM is the Mahalanobis distance between the feature vector of the object under investigation and the feature vector in the ORECIM data base. If the data base consists of w objects then after comparison, w distances will be found. The assignment is done to the minimum of these.

If the distance is above a threshold t than the object is identified as unknown.

Table 2 is a matrix showing the distances between each object in the data base used. Figure 5 gives the objects used in the laboratory.

	Obj. 1	Obj. 2	Obj. 3	Pliers	Scissor	Rect.	Star
Obj. 1	10.4	1.9E+7	2.1E+6	4.4E+9	3.6E+12	4.2E+8	6.2E+4
Obj. 2	7.5E+4	3.85	8.0E+3	1.6E+5	1.8E+8	5.6E+3	2.0E+3
Obj. 3	4.2E+10	6.5E+10	13.7	4.0E+8	3.7E+11	7.4E+5	9.5E+8
Pliers	1.1E+8	2.3E+7	1.0E+6	6.7	3.1E+6	2.9E+4	6.4E+3
Scissor	4.3E+5	2.1E+5	8.9E2	3.6E+3	4.6	3.5E+3	8.2E+2
Rect.	8.9E+9	1.1E+10	5.4E+6	3.3E+7	2.8E+10	1.6	1.6E+8
Star	3.8E+8	1.3E+8	7.4E+6	7.3E+9	6.0E+12	7.1E+8	4.2

Table2: Mahalanobis distances

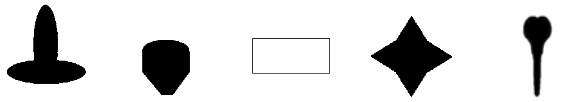

Figure 5: (Left to right) Object 1, Object 2, Rectangle, Star, Scissors

In Figure 6, the 3D plot representations of the 18 invariant vectors for each object are given. In Figure 7 the same plot with the four objects that seem close in Figure 6 is given at a different scale. The reason for this is to make clear that even if the same four object's invariants seem very close to each other in Figure 6, they are separate enough as seen in Figure 7. This closeness is because of the scale of the coordinate system.

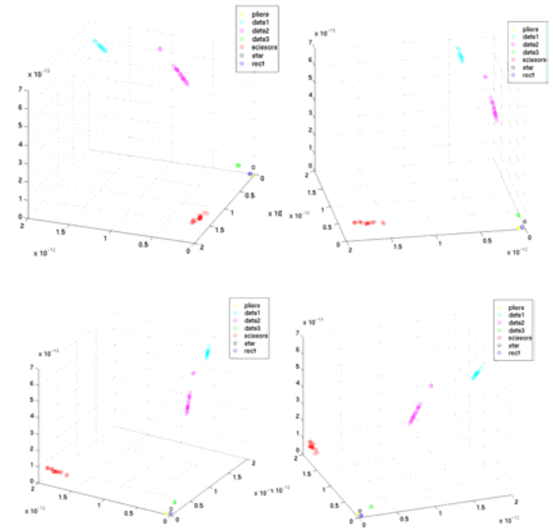

Figure 6. Three-dimensional plots from different view-angles of invariants of seven objects.

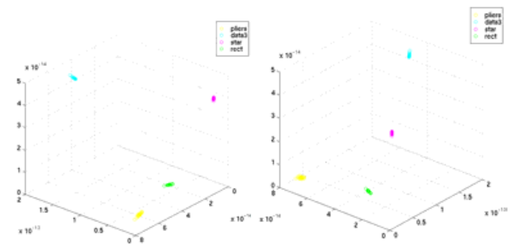

Figure 7. Three-dimensional plots of invariants of four objects, from different view-angles

I

In the second set of experiments absolute affine invariants due to Çivi, et.al are used [2]. The affine invariants used are:

$$I_1 = \frac{I_1 I_4}{I_2 I_4} \qquad I_2 = \frac{I_1 I_1}{I_3^{\,2} I_4}$$

where

$\mathbf{I}_1 = 45a_{13}^2a_{20}^2 - 30a_{12}a_{13}a_{20}a_{21} + 3a_{12}^2a_{21}^2 + 6a_{11}a_{13}a_{21}^2 + 48a_{04}a_{20}a_{21}^2 - 12a_{03}a_{21}^3 + 20a_{10}^2a_{20}a_{22} - 30a_{11}a_{13}a_{20}a_{22} - 120a_{04}a_{20}^2a_{22} - 16a_{11}a_{12}a_{21}a_{22} + 54a_{10}a_{13}a_{21}a_{22} + 12a_{03}a_{20}a_{21}a_{22} + 20a_{02}a_{21}^2a_{22} + 17a_{11}^2a_{22}^2 - 36a_{10}a_{12}a_{22}^2 - 8a_{02}a_{20}a_{22}^2 - 36a_{01}a_{21}a_{22}^2 + 72a_{00}a_{22}^3 - 12a_{12}^3a_{30} + 54a_{11}a_{12}a_{13}a_{30} - 162a_{10}a_{13}^2a_{30} - 72a_{04}a_{12}a_{20}a_{30} + 54a_{03}a_{13}a_{20}a_{30} - 72a_{04}a_{11}a_{21}a_{30} +54a_{03}a_{12}a_{21}a_{30} - 72a_{02}a_{13}a_{21}a_{30} +432a_{04}a_{10}a_{22}a_{30} - 72a_{02}a_{13}a_{21}a_{30} + 12a_{02}a_{12}a_{22}a_{30} +54a_{01}a_{13}a_{22}a_{30} -81 a_{03}^2a_{30}^2 + 216a_{02} a_{04}a_{30}^2 + 6a_{11}a_{12}^2a_{31} - 36a_{11}^2a_{13}a_{31} + 54a_{10}a_{12}a_{13}a_{31} + 180a_{043}a_{11}a_{20}a_{31}\ 72a_{03}a_{12}a_{20}a_{31} + 54a_{03}a_{12}a_{21}a_{30} - 324a_{04}a_{10}a_{21}a_{31} + 54a_{03}a_{11}a_{21}a_{31} - 30a_{02}a_{12}a_{21}a_{31} + 54a_{01}a_{13}a_{21}a_{31} + 54a_{03}a_{10}a_{22}a_{31} - 30a_{02}a_{11}a_{22}a_{31} + 54a_{01}a_{12}a_{22}a_{31} - 324a_{00}a_{13}a_{22}a_{31} + 54a_{02}a_{03}a_{30}a_{31} - 324a_{01}a_{04}a_{30}a_{31} + 45a_{02}^2a_{31}^2 - 162a_{01}a_{03}a_{31}^2+972a_{00}a_{04}a_{31}^2 - 36a_{0}\ a_{11}^2a_{40} +432a_{04}a_{10}a_{12}a_{40} - 72a_{03}a_{11}a_{12}a_{40} + 48a_{02}a_{12}^2a_{40} + 180a_{02}a_{11}a_{13}a_{40} + 432a_{04}a_{10}a_{12}a_{40} - 324a_{01}a_{12}a_{13}a_{40} + 972a_{00}a_{13}^2a_{40} + 216a_{03}^2a_{20}a_{40} - 576a_{02}a_{04}a_{20}a_{40} - 72a_{02}a_{03}a_{21}a_{40} + 432a_{01}a_{04}a_{21}a_{40} - 120a_{02}^2a_{22}a_{40} +432a_{01}a_{03}a_{22}a_{40} -2592a_{00}a_{04}a_{22}a_{40}$

$\mathbf{I}_2 = 6a_{22}a_{22}a_{22} -27a_{13}a_{22}a_{31} +81a_{04}a_{31}a_{31} +81a_{13}a_{13}a_{40} - 216a_{04}a_{22}a_{40}$

$\mathbf{I}_3 = 144a_{40}a_{04}a_{00} - 36a_{40}a_{03}a_{01} + 12a_{40}a_{02}a_{02} - 36a_{31}a_{13}a_{00} + 9a_{31}a_{12}a_{01} - 6a_{31}a_{11}a_{02} + 9a_{31}a_{10}a_{03} + 9a_{30}a_{13}a_{01} - 6a_{30}a_{12}a_{02} + 9a_{30}a_{11}a_{03} - 36a_{30}a_{10}a_{04} + 12a_{22}a_{22}a_{00} - 6a_{22}a_{21}a_{01} + 4a_{22}a_{20}a_{02} - 6a_{22}a_{12}a_{10} + 2a_{22}a_{11}a_{11} + 2a_{21}a_{21}a_{02} - 6a_{21}a_{20}a_{03} + 9a_{21}a_{13}a_{10} - a_{21}a_{12}a_{11} + 12a_{20}a_{20}a_{04} - 6a_{20}a_{13}a_{11} + 2a_{20}a_{12}a_{12}$

$\mathbf{I}_4 = 120a_{40}a_{04} - 30a_{31}a_{13} + 10a_{22}a_{22}$

The objects used in the experiments are shown in Figure 8.

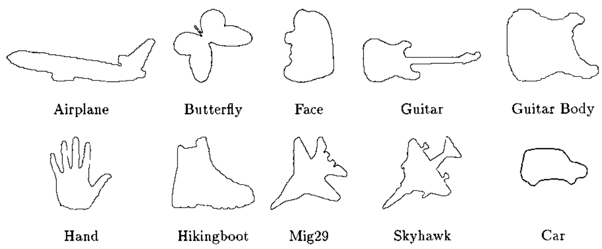

Figure 8. Objects used in the experiments

For each shape, random affine transformations are generated and using the 3L fitting algorithm, 4th degree implicit polynomials are fitted to the tranformed data sets. For each shape 350 such transformations and the resulting invariant space representations are included. Figure 9 shows at two different scales, the manifolds generated by some of the shapes in the invariant space spanned by $\mathbf{I}_1$ and $\mathbf{I}_2$ Note that ideally, each shape should correspond to a single point in the invariant space. However, the 3L fitting is not fully affine invariant, and in the invariant space the resulting error maps the shape representation into clouds rather than to a unique point.

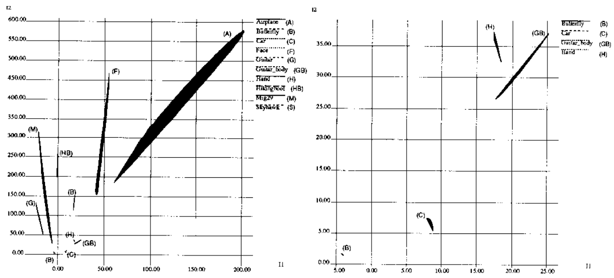

Figure 9. Invariants of the objects in Figure 8.

The results indicate a clear separation of the shape boundaries in the invariant space.

2.4. Parametric-Implicit Conversion

Many current CAD systems use parametric forms to represent and store object data (or models). In order to use these representations in ORECIM, parametric forms should be converted to their implicit counterparts using an appropriate conversion method. This would allow ORECIM to store object data coming from the CAD system and use it for object recognition.

There are three known techniques for implicitization [22]. The first technique involves the use of Elimination theory. The second technique utilizes Gröbner bases. It computes a canonical representation of the ideal generated by the parametric equations, by defining a suitable ordering of the variables. This technique is fairly expensive in practice and even for low degree parameterizations it may take a lot of time.

The third one, called Wu-Ritt's Method, transforms the given parametric polynomials until it has a certain form, whereupon the question about f is answered. The transformation involves rewriting the polynomials in F using pseudo-division and adding the remainders to the set F. f is the polynomial that encodes the conclusion of the theorem (corresponding implicit form) and F is the set of polynomials that encode the hypothesis (the set of parametric equations).

An Elimination theory based method called "Cayley's Method" is adequate for 2D rational parametric curves and easy to understand and implement in computer environment.

Example: The following 2D parametric equations

$$x = \frac{t^4 - 52t^3 + 63t^2 - 15t + 7}{-37t^2 + 3t + 1}$$
$$y = \frac{4}{-37t^2 + 3t + 1}$$

are converted into the implicit correspondent

$$1296x^2y^2 - 828xy^3 - 18143y^4 - 10656xy^2 + 169076y^3 - 1152xy - 470928y^2 + 484736y + 256$$

applying Cayley's Method. Figure 6 shows the plots of both parametric and implicit representations for $t \in [0,1]$.

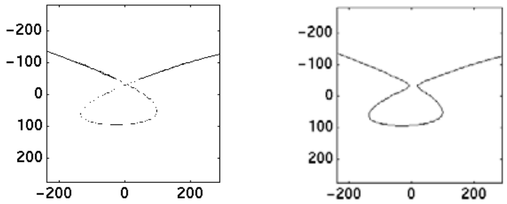

Figure 6: Parametric form plot (left), implicit form plot (right)

After some trials it is noted that the invariant vectors computed from the implicit polynomial converted from a parametric form and the from the fitted implicit polynomial to the data set generated by the same parametric form are not the same. This is probably due to the fact that even if both converted and fitted implicit polynomials look the same locally, they differ globally. Since invariants depend on the coefficients, they are global descriptors.

3. CONCLUSION

In this paper a recognition system based on fitting an implicit polynomial to the data of an unknown object and using its algebraic invariants for the identification of the object is developed. The results obtained running it on a limited number of images show success. This fact highlights the importance of the implicit polynomials as a tool that can be applied for solving industrial problems

Even if the system is not tested on a real production system, the performance under laboratory environment encourages its implementation.

The conversion of parametric representations of objects to implicit ones is also studied. Even if successful conversions were obtained, the proposed methods were not able to find any relation between the converted implicit form and fitted one of the same object. This can be suggested as a further research since it would make possible the integration of the CAD and vision systems.

Acknowledgements: This research has been partially supported by DPT 95K120320, BU Research Fund 94A0312 and Tübitak MISAG-66.

References

[1] Blane, M.M., Lei, Z. and Cooper, D.B., 'The 3L Algorithm for fitting implicit polynomial curves and surfaces to data', *Brown University LEMS Lab. Technical Report,* No. 160, Feb 1997

[2] Çivi, H., Christopher, C., Erçil, A., 'The Classical Theory of Invariants and Object Recognition using Algebraic Curves and Surfaces', *Boğaziçi University Research Report*, FBE-IE-06/97-07

[3] Erdem, U.M. "2D Object Recognition Using Implicit Polynomials and Algebraic Invariance" *Masters Thesis,* Boğaziçi University, Turkey 1996.

[4] Hoffmann C. M., "Advanced Curve and Surface Techniques", *Eurographics '93,* Barcelona, September, 1993.

[5] Keren D., "Using Symbolic Computation to Find Algebraic Invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 1143-1149, November 1994.

[6] Manocha D. and Canny J.F., "Implicit Representation of Rational Parametric Surfaces", *Journal of Symbolic Computation*, Vol. 13, pp. 485-510, 1992.

[7] Sederberg T. W., Anderson D. C. and Goldman R. N., "Implicit Representation of Parametric Curves and Surfaces" in *Computer Vision, Graphics and Image Processing*, pp. 72-84, Academic Press, 1984.

[8] Subrahmonia J., Cooper D. B. and Keren D., Practical Reliable Recognition of 2D and 3D Objects Using Implicit Polynomials and Algebraic Invariants, , *Brown University LEMS Lab. Technical Report* No-107, June, 1992.

[9] Taubin G. and Cooper D. B., Recognition and Positioning of 3D Piecewise Algebraic Objects Using Euclidean Invariants, IBM Research Division, T.J. Watson Research Center, 1990.