2D OBJECT DESCRIPTION AND RECOGNITION BASED ON CONTOUR MATCHING BY IMPLICIT POLYNOMIALS

Zoya Landa, David Malah, and Meir Barzohar

Department of Electrical Engineering, Technion- Israel Institute of Technology, Haifa 32000, Israel.
phone: +(972)-4-8294745, fax: +(972)-4-8292793,
email: zrybak@yahoo.com, malah@ee.technion.ac.il, meirb@visionsense.com.

ABSTRACT

This work deals with 2D object description and recognition based on coefficients of implicit polynomials (IP). We first improve the description abilities of recently published Min-Max and Min-Var algorithms by replacing algebraic distances by geometric ones in the relevant cost function. We propose a new recognition approach that is based on deriving linear rotation invariants from several polynomials of different degrees, fitted to the object shape, as well as on their fitting errors. This approach is found to considerably improve the recognition and is denoted as Multi Order (degree) and Fitting Errors Technique (MOFET). We also use a Shape Transform, based on the Scatter Matrix of the objects' shape, to allow Affine invariant classification. Finally, we compare the performance of our approach with the Curvature Scale Space (CSS) method and find that it has an advantage over CSS, at about the same complexity.

1. INTRODUCTION

Implicit polynomials have been exploited for quite some time for the representation and recognition of 2D data, defined by discrete points taken along its boundary [1] – [8], [12]-[14]. Over the years, different algorithms for fitting Implicit Polynomials to the data were developed. Early iterative fitting algorithms, like [4] often failed to derive an Implicit Polynomial defining a single closed curve that fits well the boundary of the data, i.e., they were unable to obtain a reliable representation. Furthermore, due to numerical instability and high computational cost, these algorithms are able to fit the data by polynomials of relatively low degree only (usually, up to degree 4), and, therefore, fail to represent or recognize relatively complicated objects. Recent versions of fitting algorithms: 3L [1] and, especially, Gradient1 [2], Min-Max and Min-Var [3], apply a linear LS (Least Squares) solution to the fitting problem (having, therefore, a lower computational cost than earlier iterative algorithms), appear to have much better performance in both representation and recognition tasks. Hence, in our work we focus on the last three algorithms, namely: Gradient1, Min-Max and Min-Var.

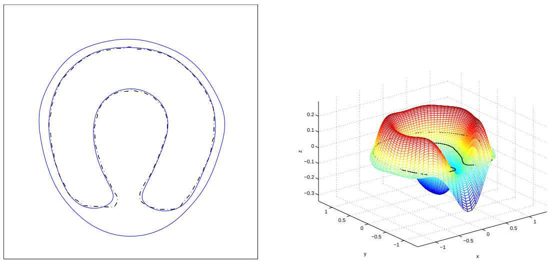
2. BACKGROUND

The ability to efficiently describe curves that represent the boundary of 2D objects is important in computer vision tasks and computer graphics. Implicit Polynomials provide a solution to this problem, using the coefficients of the polynomial to represent the data. The technique assumes that the data is a curve that is lying in the zero-plane (i.e., in 3D space, with axes x,y,z , the zero-plane is defined by z=0), and that one of the zero-sets of the polynomial (z= f(x,y)) is supposed to fit this curve.

An example of a curve lying in the zero-plane and a polynomial describing it is shown in Fig. 1.

2.1 2D Object description by Implicit Polynomials

As will be shown in the sequel (subsection 2.2), a polynomial that efficiently describes a 2D object, given by a set of points (data-set) along the object's boundary, should meet several conditions. We will start here from the obvious and most intuitive requirement: the zero-set of the fitting polynomial should be close to the data-set.



(a) (b)

Figure 1: (a) - curve representing the boundary of an object (dashed line) and the zero-sets of the 4-th degree polynomial attempting to fit the object (solid). Note that there is a spurious zeros-set (outside solid curve). (b) - surface of a 4-th degree polynomial attempting to fit the object.

Let's define:

$\underline{x} \triangleq [x_1 \;\; x_2 \; \dots \; x_n]'$: a vector of the first coordinates of the data set, where $(\cdot)'$ denotes the transpose of $(\cdot)$;

$\underline{y} \triangleq [y_1 \;\; y_2 \; \dots \; y_n]'$: a vector of the second coordinates of the data set;

$P^r_{\underline{a}}(x,y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 + \ldots + a_{r,0}x^r + a_{r-1,1}x^{r-1}y + \ldots + a_{0,r}y^r$: a polynomial of degree r with coefficients-vector

$$\underline{a} \triangleq [a_{0,0} \;\; a_{1,0} \;\; a_{0,1} \; \dots \; a_{r,0} \;\; a_{r-1,1} \; \dots a_{0,r}]'. \tag{1}$$

The above expression can then be written as:

$$P^r_{\underline{a}}(x,y) = \underline{a}'\underline{p}^r(x,y), \tag{2}$$

where

$$\underline{p}^r(x,y) \triangleq [1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2 \; \dots \; x^r \;\; x^{r-1}y \; \dots \; y^r]' \tag{3}$$

is a vector of monomials at a point (x,y).

The length of both the monomial vector and the coefficient vector is $\frac{(r+1)(r+2)}{2}$. The zero-sets of $P^r_{\underline{a}}(x,y)$ are denoted by $ZS\left(P^r_{\underline{a}}(x,y)\right) = \left\{(x,y) : P^r_{\underline{a}}(x,y) = 0\right\}$.

Note, that a polynomial may have several zero-sets, where a zero-set is defined as a continuous curve satisfying $P^r_{\underline{a}}(x,y) = 0$. Actually, a polynomial of degree r may have up to r zero-sets.

Thus, we would like to find a polynomial having a zero-set that best fits the given data-set for different criteria (equations (5), (6) and (12)).

2.2 Gradient-one, MinMax and MinVar algorithms

In this subsection we bring a brief overview of the recent fitting algorithms that are used in the sequel. These algorithms are based on a simple linear LS solution, generally produce a single zero-set that is close to the data-set (instead of two or more zero-sets fitting the data-set), and are numerically stable even when using polynomials of relatively high degree (in our work we used up to an 8-th degree polynomials).

1. **Gradient-one algorithm**

In order to produce a single zero-set resembling the data-set, the Gradient-one algorithm [2] attempts to fulfill two conditions. The first condition is to minimize the algebraic distance between the polynomial and the data-set: i.e.,

$$\min_{\underline{a}} \sum_{k=1}^{n} P_{\underline{a}}^{r}(x_k,y_k)^2 \tag{4}$$

The second condition exploits the fact that the gradient vector at a point belonging to the zero-set is perpendicular to the zero-set tangent at this point. Assuming that the resulting zero-set indeed lies nearby the data-set, the Gradient-one algorithm requires that the gradient of the polynomial at the data-set points will be perpendicular to the data-set curve-tangent. In addition, the authors of [2] show that large fluctuations in the first derivative of the polynomial can result in a polynomial sensitive to data-set fluctuations. Hence, the required value of the gradient at data-set points is set to 1.

Thus, the Gradient-one algorithm formulates the following minimization problem:

$$\begin{aligned} \min_{\underline{a}}\{ \quad & \textstyle\sum_{k=1}^{n} P_{\underline{a}}^{r}(x_k,y_k)^2 + \\ & \textstyle\sum_{k=1}^{n} (\underline{g}_N'(x_k,y_k)\cdot\underline{a}-1)^2 + \\ & \textstyle\sum_{k=1}^{n} (\underline{g}_T'(x_k,y_k)\cdot\underline{a})^2 \}. \end{aligned} \quad , \tag{5}$$

where $\underline{g}_N(x,y)$ and $\underline{g}_T(x,y)$ are derivatives of monomials at normal and tangent directions to the data-set, respectively.

2. **Min-Max and Min-Var Algorithms**

These two algorithms [3] aim to reduce the sensitivity of the zero-set to polynomial coefficients perturbations. The algorithms propose to constrain the values of the gradient at the data-set points (which are assumed to be close to the derived zero-set), so that all points belonging to the obtained zero-set would be equally sensitive to small coefficient perturbations. As a result, the algorithms propose to minimize:

$$\begin{aligned} \min_{\underline{a}} \quad & \{\textstyle\sum_{i=1}^{n} P_{\underline{a}}^{r}(x_i,y_i)^2 + \\ & \textstyle\sum_{i=1}^{n} \left(\frac{\underline{g}_T'(x_i,y_i)\cdot\underline{a}}{w_{m(m/v)}(x_i,y_i)}\right)^2 + \\ & \textstyle\sum_{i=1}^{n} \left(\frac{\underline{g}_N'(x_i,y_i)\cdot\underline{a}}{w_{m(m/v)}(x_i,y_i)}-1\right)^2 \} \end{aligned} \tag{6}$$

where $w_{m(m/v)(x,y)}$ means $w_{mm}(x,y)$ *or* $w_{mv}(x,y)$, for Min-Max and Min-Var, respectively.

$$w_{mm}(x,y) \stackrel{\triangle}{=} \sum_{i=1}^{\frac{(r+1)(r+2)}{2}} |p_i^r(x,y)| \tag{7}$$

and

$$w_{mv}(x,y) \stackrel{\triangle}{=} \sqrt{\underline{p}^r(x,y)'\underline{p}^r(x,y)}. \tag{8}$$

3. MODIFICATION OF MIN-MAX AND MIN-VAR ALGORITHMS

3.1 Improving Fitting Performance

Min-Var and Min-Max algorithms minimize the polynomial value along the data-set. We propose to minimize the approximated fitting error (see [4]):

$$\min_{\underline{a}} \sum_{k=1}^{n} \left(\frac{P_{\underline{a}}(x_k,y_k)}{||\bigtriangledown P_{\underline{a}}(x_k,y_k)||}\right)^2, \tag{9}$$

instead.

Naturally, the value of the gradient at every point of the data-set is needed. Although the exact value of the gradient is not available before the problem is solved, in the case of Min-Max and Min-Var we have a clue of its value. In the case of Min-Max a good solution presupposes values of the gradient at a point (x,y) to be close to $w_{mm}(x,y)$ (7), and in the case of Min-Var, close to $w_{mv}(x,y)$ (8). Thus, substituting these values into (9) we get:

$$\min_{\underline{a}} \sum_{i=1}^{n} \left(\frac{P_{\underline{a}}(x_i,y_i)}{w_{m(m/v)}(x_i,y_i)}\right)^2. \tag{10}$$

However, by dividing the first element in (6) by a factor bigger than one, we weaken its influence on a problem solution. To regain its weight we multiply (10) by

$$W_{m(m/v)} = \frac{1}{n}\sum_{i=1}^{n} w_{m(m/v)}(x_i,y_i) \tag{11}$$

and thus get the following minimization problem:

$$\min_{\underline{a}} \left\{ \sum_{i=1}^{n} \left(W_{m(m/v)}\frac{P_{\underline{a}}(x_i,y_i)}{w_{m(m/v)}(x_i,y_i)}\right)^2 + \sum_{i=1}^{n} \left(\frac{\underline{g}_T'(x_i,y_i)\cdot\underline{a}}{w_{m(m/v)}(x_i,y_i)}\right)^2 + \sum_{i=1}^{n} \left(\frac{\underline{g}_N'(x_i,y_i)\cdot\underline{a}}{w_{m(m/v)}(x_i,y_i)}-1\right)^2 \right\} \tag{12}$$

Figures 2 and 3 demonstrate the difference in fitting between the original Min-Max/Min-Var and the modified Min-Max/Min-Var (Mod. Min-Max/ Mod. Min-Var). If we compare the fittings, we conclude that the original algorithms make more effort to fit points lying far away from the origin, than points lying close to it; while the modified algorithms result in a more uniform solution (further details can be found in [19]).

3.2 Invariance to rotation

Although, Min-Max and Min-Var algorithms show better fitting performance than Gradient-one, these algorithms are not rotation invariant (which is important in recognition tasks), due to the $w_{m(m/v)}$ factor. We propose to substitute it by a factor dependent only on the distance of the data-set points from the origin (and update $W_{m(m/v)}$, respectively), thus gaining rotation invariance. In the case of Min-Max, (7) is substituted by

$$w_{mm}^{ri}(d_k) = \underset{x,y,\ x^2+y^2=d_k^2}{mean}\left(w_{mm}(x,y)\right). \tag{13}$$

The use of this average value is not expected to harm the Min-Max or Mod. Min-Max properties. Fig. 4 shows the value of $w_{mm}^{ri}(d)$ as a function of d along with the minimum and the maximum values of $w_{mm}(x,y)|_{x^2+y^2=d^2}$. It can be seen that for high degree polynomials these three quantities hardly differ.

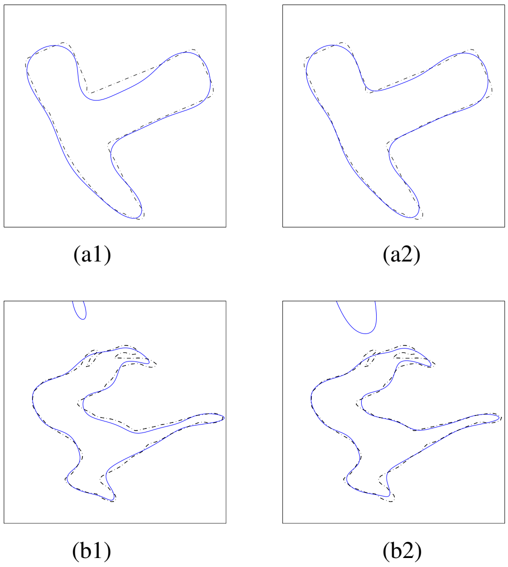


Figure 2: Min-Max (left) and Mod. Min-Max (right) fit (polynomial of degree 8). Data-set – dash-dotted line, obtained zero-sets – solid.

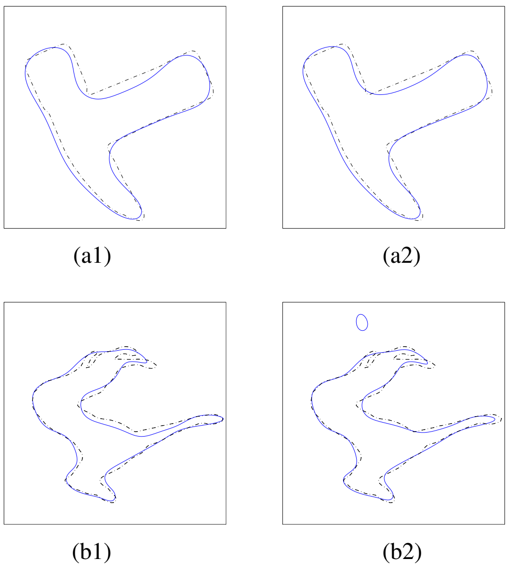


Figure 3: Min-Var (left) and Mod. Min-Var (right) fit (polynomial of degree 8). Data-set – dash-dotted line, obtained zero-sets – solid.

In the case of Min-Var we redefine the polynomial as follows:

$$\begin{aligned} P_{\underline{a}}(x,y) &= \textstyle\sum_{l=0}^{r}\sum_{i=0}^{l} a_{l-i,j}x^{l-i}y^{i} \\ &= \textstyle\sum_{l=0}^{r}\sum_{i=0}^{l} \underbrace{\frac{a_{l-i,i}}{\sqrt{\binom{l}{i}}}}_{b_{l-i,i}} \cdot \sqrt{\binom{l}{i}}x^{l-i}y^{i} \\ &= \textstyle\sum_{l=0}^{r}\sum_{i=0}^{l} b_{l-i,i}\sqrt{\binom{l}{i}}x^{l-i}y^{i} \\ &\triangleq \mathscr{P}_{\underline{b}}^{r}(x,y), \end{aligned} \tag{14}$$

where $\mathscr{P}_{\underline{b}}^{r}(x,y)$ denotes a polynomial of degree r, and the vector $\underline{b}$ is comprised of coefficients $b_{l-i,i}$ that multiply the *factorized* monomials of the form $\sqrt{\binom{l}{i}}x^{l-i}y^{i}$. Thus, (8) rewritten with respect

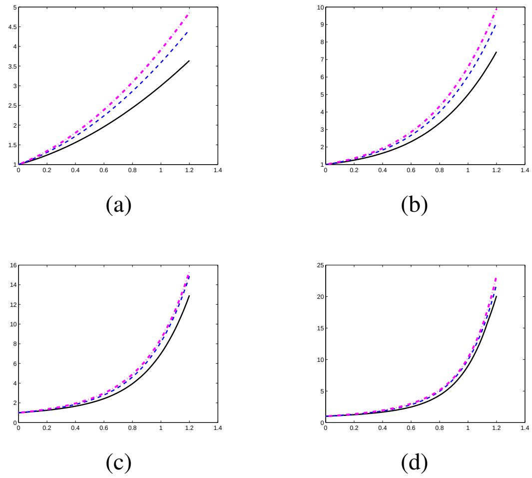


Figure 4: $w_{mm}^{ri}(d)$ (dashed line) along with the minimum (solid line) and the maximum (dash-dotted line) value of $w_{mm}(x,y)|_{x^2+y^2=d^2}$ as a function of the distance from the origin d for different degrees of the polynomial. (a) 2-nd degree, (b) 4-th degree, (c) 6-th degree, (d) 8-th degree.

to vector of *factorized* monomials (which is sensitive to $b_{l-i,i}$), becomes

$$\begin{aligned} w_{mv}^{ri}(x,y) &\triangleq \sqrt{\textstyle\sum_{l=0}^{r}\sum_{i}^{l}\binom{l}{i}x^{2\cdot(l-i)}y^{2i}} \\ &= \sqrt{\textstyle\sum_{l=0}^{r}(x^2+y^2)^{l}} \end{aligned} \tag{15}$$

Note also, that because the data-set lies mostly inside a circle of radius 1, the higher the monomial power the bigger its coefficient. On the other hand, the factors, $\sqrt{\binom{l}{i}}$ also tend to grow with monomials power. Thus the dynamic range of $\underline{b}$ is smaller than the dynamic range of $\underline{a}$ and, hence, $\underline{b}$ is less sensitive to coefficient noise, as it can be seen in Fig. 5.

4. APPLICATION OF IMPLICIT POLYNOMIALS TO 2D OBJECT RECOGNITION

The ability to recognize objects is very important in many fields, such as medical and robotic fields. In this section we propose a novel IP based technique.

4.1 Data preprocessing

In our work we took some preprocessing steps, before applying the IP algorithm on data. First, we filtered out the measurement noise by passing the data-set through a low-pass filter (the procedure appears in [3]). Then, in order to provide affine invariant classification, the data was centered and its Scatter Matrix normalized. The use of this concept in combination with different recognition algorithms can be found, for example, in [10] and [11]. If there is no need in affine invariant classification, this step can be easily skipped. Finally, we put the data-set close to the unit circle in order to avoid large perturbations in the coefficients as a result of data-set noise (see [2]). We center the data-set and then apply $\mathbb{S}_{75\%}$ scaling factor that scales the data-set so that 75% of its points would appear inside the unit circle and the rest – outside of it. This scaling factor ia stable to outliers, and provide good distinguishing between different shapes.

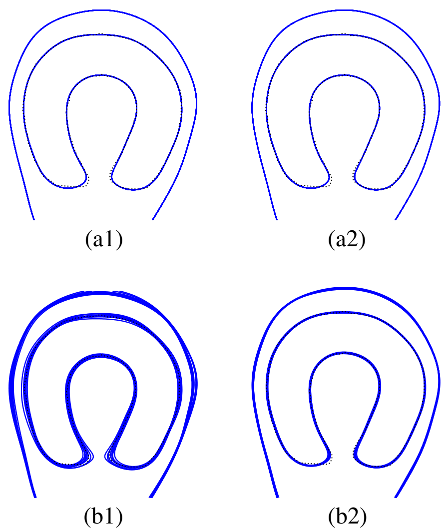


Figure 5: Left column: Application of the original Mod. Min-Var of degree 8 to the "Horseshoe" shape. Right column: Application of the RI Mod. Min-Var of degree 8 to the "Horseshoe" shape. (a) Heavy-dotted line: the original data-set, solid line – the fitted polynomial zero-set. (b) Heavy-dotted line: the original data-set. Solid lines – accumulated zero-sets, when noise added to each coefficient is uniformly distributed with mean 0 and absolute maximum value of 2^{-10} of the biggest coefficient.

4.2 Multi Order (degree) and Fitting Error Technique (MOFET) for data-set recognition

4.2.1 Drawbacks of single-degree recognition

All the aforementioned algorithms use coefficients of a polynomial of a predefined degree (typically, 6 or 8) to recognize objects. However, generally, the dictionary would contain contours of different complexity. Thus, we can find some contours that cannot be fitted well by the polynomial of the chosen degree, and, as a result, the resulting zero-set may resemble other shapes in the dictionary, making them indistinguishable. On the other hand, we might have some simple contours that can be fitted by a variety of polynomials.

4.2.2 Recognition based on several polynomials

We propose the following solution to the single degree recognition problem: *Matching polynomials of several different degrees to each of the data-sets*.

This way we make sure that for each object there exists a set of reliable and valuable parameters. In our work we matched polynomials of degree 2, 4, 6 and 8. We calculated the linear geometric invariants [7] from the obtained polynomial coefficients. Thus, we got a relatively small (15 features) and stable dictionary. In order to improve the classification, we also use fitting error feature. That is, for each fit, we calculate a vector of fitting errors $\underline{e_a}$ and then choose as a feature the value that is greater than 75% of the elements, but smaller than the rest 25%. This way we get four more (for the four polynomial degrees) features - a total of 19 features for each fit. We name the proposed technique which takes advantage of both the coefficients of fitting polynomials of different degrees and fitting errors, Multi Order and Fitting Error Technique (MOFET).

5. EXPERIMENTS

5.1 Recognition process

Generally, a recognition task consists of two sub-tasks: Recognizer design (i.e., learning) and Recognizer application (i.e., testing). In our work it is assumed that the feature vectors from each set of objects have a Gaussian probability density function (PDF) of feature vectors, where the parameters of each PDF (i.e., the mean vector and covariance matrix) are estimated from feature vectors belonging to the different views (after data preprocessing as described in subsection 4.1) of a corresponding object in the dictionary. Thus, each dictionary object is represented by its PDF. Then, we apply the Maximum Likelihood principle for recognition. I.e., we extract the feature vector from the shape we want to recognize and substitute it into each object's PDF. We then recognize the shape as the object that got the highest likelihood value.

5.2 Experimental Database

In our experiments we used the "Multiview Curve Database" (MCD) created by M. Zuliani [18], containing 40 different shapes taken from 7 different points of view, giving a total of 280 basic shapes. Examples of different views of the same object are shown in Fig. 6a. In order to expand this data-base, we added colored noise to the coordinates of each shape. The colored noise was generated as follows: normally distributed random noise with std of 0.02 was filtered by a Gaussian window of a width (std) of 2.5. The filter's coefficients were normalized such that the sum of squares of its coefficients equals 1. This procedure was carried out twice for each perturbation: for x and y coordinates of the shape. Each basic shape was perturbed 20 times by adding colored noise, thus a database of $40 \times 7 \times 20 = 5600$ different objects was created. An example of the effect of these perturbations appear in Fig 6b. Note that because of the shape Scater Matrix normalization the "stretched" data-sets are more sensitive to noise then those which undergo a minor normalization.

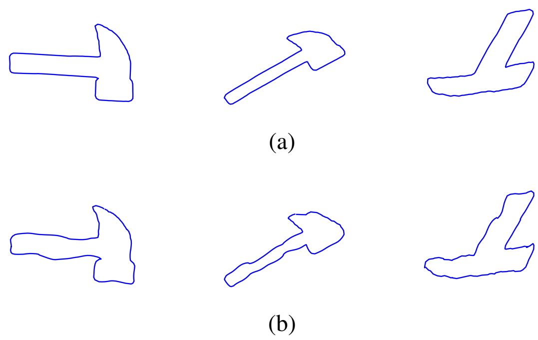


Figure 6: (a) 3 different views of the shape "Hammer", (b) The views perturbed by colored noise with STD = 0.02

5.3 Experimental Results

In our experiments we chose a group of objects from the data-set obtained in section 5.2 for the recognizer dictionary (i.e., learning data), while the rest (test data) were recognized using the obtained dictionary. For learning we used 6 views $\times$ 20 perturbations = 120 data-sets for each of the 40 shapes in the database. The remaining sets (20 data-sets for each shape) were used for the recognition test. Then, the MOFET algorithm was applied and the recognition rate (i.e., the percentage of correctly recognized shapes) was calculated.

In order to get more statistics, we carried out the above process several times, where each time different views were used for learning. Then, the average recognition rate was calculated.

The results of this process for the different recognition algorithms examined are shown in Table 1

As can be seen, there is only a slight difference in performance between the different examined algorithms, when applied with MOFET. We choose Mod. RI-Min-Var for other experiments.

We compared the MOFET approach to an algorithm that uses only a single-degree along with the ridge-regression factor providing the best classification rate (see [2]) (i.e., using invariants of a

Table 1: Performance of MOFET for different fitting algorithms.

Grad.One	Mod. RI-Min-Max	Mod. RI-Min-Var
96.5%	96.2%	**96.6%**

single fitting polynomial). The results are shown in Table 2, proving the effectiveness of the MOFET approach. Note that MOFET doesn't require adjustment of the ridge-regression factor.

We also performed experiments with 5% missing data. First, we used 6 views $\times$ 20 perturbations = 120 data-sets for each of the 40 shapes in the database. The remaining set (20 perturbations for each shape) was randomly occluded and used for the recognition test. As a result we got 91% recognition rate. Then, we produced a more complete dictionary, based on all the 7 views (without occlusion). For the recognition step, the data-sets were perturbed again and randomly occluded. As a result we got 95% recognition rate.

Table 2: Comparison between MOFET and single-degree approaches.

MOFET	Single-degree			
	2	4	6	8
96.6%	69.4%	**94.8%**	94.2%	85.8%

Finally, we compared the performance of MOFET with the CSS [15] technique (again, 6 views were used for learning). We chose this technique, because is has been reported [15] as having better performance then other known techniques, such as Fourier Descriptors [16] and Convex Hull [17]. We also compared the techniques in a less noisy environment, i.e., added colored noise of STD = 0.01, which resulted in only slight data-set deformations. The results of the comparisons appear in Table 3. It can be seen that the MOFET algorithm has better performance at both noise levels examinated.

Table 3: Comparison between MOFET and CSS techniques.

STD = 0.02		STD = 0.01	
MOFET	CSS	MOFET	CSS
96.6%	93.8%	**99.6%**	97.5%

6. CONCLUSION

In this work we have improved the performance of existing fitting algorithms in two aspects: representation and recognition. We have shown that replacing the algebraic distances by geometric ones, in the fitting algorithm minimization problem, results in improved fitting of the data-set.

We have also introduced a novel Multi Order (degree) and Fitting Error (MOFET) recognizer that outperforms existing Implicit Polynomial based recognizers. Fitting several polynomials of different degrees and utilization of their coefficients along with the fitting errors made it possible to take advantage both of the stability of low-degree polynomials and informativeness of high-degree polynomials, and thus enabled the design of a high performance recognizer. Making use of the linear invariants designed by Tarel [7], allows the recognition of objects that underwent an Euclidian transform. Exploiting the Scatter Matrix normalization technique enables effective projection-based recognition of 3D objects in a circumstances allowing approximations of a projective transform by an Affine transform.

Finally, we compared the proposed MOFET recognizer to the standard CSS [15] contour based recognition and found MOFET to have a better performance.

REFERENCES

[1] M. M. Blane, Zhibin Lei, “3L Algorithm for Fitting Implicit Polynomial Curves and Surfaces to Data”, IEEE Trans. on PAMI, vol.22, No. 3, pp. 298–313, March 2000.

[2] T. Tasdizen, J.P. Tarel and D.B. Cooper, “Improving the Stability of Algebraic Curves for Application”, IEEE Trans. on Image Proc., vol.9, No. 3, pp. 405–416, March 2000.

[3] A. Helzer, M. Barzohar and D. Malah, “Robust Fitting of 2D Curves and 3D Surfaces by Implicit Polynoms”, IEEE Trans. on PAMI., vol. 26, no. 10, pp. 1283–1294, Oct. 2004.

[4] G. Taubin, ”Estimation of Planar Curves, Surfaces and Nonplanar Space Curves Defined by Implicit Equations, with Applications to Edge and Range Image Segmentation”, Trans. on PAMI, Vol. 13, No. 11, pp. 1115-1138, Nov. 1991.

[5] D Keren, C Gotsman, ”Fitting curves and surfaces with constrained implicit polynomials ”, IEEE Trans. PAMI, vol. 21, pp. 31-41, Jan. 1999.

[6] C. Unsalan, ”A model based approach for pose estimation and rotation invariant object matching”, Pattern Recognition Letters 28 (2007), pp. 49-57.

[7] J.-P. Tarel, D.B. Cooper, ”A New Complex Basis for Implicit Polynomial Curves and its Simple Exploitation for Pose Estimation and Invariant Recognition”, CVPR, p. 111, 1998.

[8] J.-P. Tarel, W. A. Wolovich, D.B. Cooper, ”Covariant-Conics Decomposition of Quartics for 2D Shape Recognition and Alignment”, Journal of Mathematical Imaging and Vision archive Volume 19 , Issue 3, pp. 255 - 273, Nov. 2003.

[9] A. Goshtasby, “Description and Discrimination of Planar Shapes Using Shape Matrices ”. IEEE Trans. on PAMI, vol. 7, pp.738-743, Nov. 1985.

[10] Y. Avrithis, Y. Xirouhakis, S. Kollias, ”Affine-invariant curve normalization for object shape representation, classification, and retrieval”, Machine Vision and Applications, vol. 13, no. 2, pp. 80-94, Nov. 2001.

[11] M. Zuliani, S. Bhagavathy, B. S. Manjunath, C. S. Kenney, ”Affine-Invariant Curve Matching” IEEE Int. Conf. on Image Processing, Singapore, Oct. 2004.

[12] J. Subrahmonia, D. B. Cooper, D. Keren ”Practical reliable Bayesian recognition of 2D and 3D objects usingimplicit polynomials and algebraic invariants”, IEEE Trans. on PAMI, vol. 18, Issue 5, pp. 505-519, May 1996.

[13] L. Zhibin, T. Tasdizen, D. B. Cooper, ”PIMs and invariant parts for shape recognition,” in Proc. 6th Int. Conf. Computer Vision (ICCV’98), Mumbai, India, 1998, pp. 827-832.

[14] M. Barzohar, D. Keren, D. B. Cooper, ”Recognizing Groups of Curves Based on New Affine Mutual Geometric Invariants, with Applications to Recognizing Intersecting Roads in Aerial Images”, IAPR Int. Conf. Pattern Recognition, vol. 1, pp. 205-209, Oct. 1994.

[15] F. Mokhtarian, S. Abbasi, and J. Kittler, ”Robust and Efficient Shape Indexing through Curvature Scale Space”, In BMCV, Edinburgh, UK, 1996.

[16] F. Chaker, M.T. Bannour, F. Ghorbel, ”A complete and stable set of affine-invariant Fourier descriptors”, Proc. of Int. Conf. on Image Analysis and Processing, pp. 578-581, Mantova, Italy (2003).

[17] Z. Yang, F. S. Cohen, ”Image registration and object recognition using affine invariants and convex hulls”, IEEE Trans. on PAMI, vol. 8, pp. 934-946, July 1999.

[18] http://vision.ece.ucsb.edu/∼zuliani/MCD/MCD.shtml

[19] Z. Landa ”2D object description and classification based on contour matching by implicit polynomials”, M.Sc Thesis, Technion Israel Institute of Technology, Haifa, Israel, 2006. Available at: http://sipl.technion.ac.il/siglib/FP/Zoya_Landa.pdf