AFFINE INVARIANT FITTING OF IMPLICIT POLYNOMIALS

* Hakan Civi ** Aytül ERÇİL
*AT Kearny, Istanbul Turkey
**Sabancı University, Engineering and Natural Sciences
aytulercil@sabanciuniv.edu

Abstract

Combining implicit polynomials and algebraic invariants for representing and recognizing complicated objects proves to be a powerful technique. However, a basic requirement for using implicit polynomials for affine invariant recognition is to have an *affine invariant fitting algorithm*. In this paper, we study the problem of affine invariant fitting of an implicit polynomial to data.

1-INTRODUCTION

There are many techniques available to describe object boundaries, such as B-Splines, Non-uniform Rational B-Splines, Fourier descriptors, chain codes, polygonal approximation, curvature primal sketch, medial axis transform, however, most of them are not appropriate for invariant object recognition. Implicit Polynomials are potentially among the most useful object representations for object recognition due to the existence of invariants under certain transformations.[2-6]

Invariants can be defined as quantities assigned to objects that do not change when the coordinate system undergoes transformations, and hence are good descriptors for recognition. For instance, if you deal with shapes described solely by ellipsoids and the transformations are Euclidean, the lengths of the various axes of the ellipsoids are invariants. If, however, affine or projective transformations are allowed, the axes are no longer invariants.

Affine transformations have been extensively studied in computer vision and, within implicit polynomial paradigm. However, if affine properties of implicit polynomials, such as affine invariants or pose estimators, are to be employed to determine the affine equivalence between two curves, a basic requirement is to have an *affine invariant fitting algorithm*.

In this paper, we study the problem of affine invariant fitting of an implicit polynomial to data.

In Section 2, we discuss the problem of fitting implicit polynomials to data, in Section 3, we propose an affine invariant fitting techniques for implicit polynomials. Section 4 details the experiments carried out for assessing the performance of the proposed technique.

2. Implicit Polynomial Fitting

An essential requirement for practicality of implicit polynomial techniques is to have robust and consistent implicit polynomial fits to data sets. The fitting problem of implicit polynomials is to find the implicit polynomial function f(x,y) that best represents the object. Since the implicit polynomial coefficients we find from the fittings are used to compute algebraic invariants or conic factor centers, which are used as primary geometric descriptors of the objects modeled in the recognition system, the fitting procedure has to be robust. This means a small amount of change in the data should not cause a relatively huge amount of change in the coefficients of the fitted implicit polynomial. In that sense, some researchers claim that the stability of the fitting is more important than the goodness of the fitting itself. In our work, we used the 3L fitting technique, which has a good record of robustness when compared with other methods in the literature. [3- 5] 3L fitting uses a pair of level sets Γ_{+c} and Γ_{-c} , where both level set are at a distance +c (outside the data set) and -c (inside the data set) from the data set, respectively. [1] Besides the original data set, the 3L algorithm uses a pair of synthetically generated data sets consisting of points from Γ_{+c} and Γ_{-c}. This procedure forces the solution to be more stable around the data. The 3L fitting algorithm is inherently Euclidean but not affine invariant. The reason for this is that the level set generation is bad on a Euclidean invariant quantity, i.e., the distance of two points, which is not preserved under an affine transformation. As a result, after affine transformations of data sets the level sets generated by the Euclidean distance transform on the transformed data set are not identical to the same affine transformation of the corresponding level sets for the original data set.

3. Affine Invariant 3L fitting

Given a data set and an affine transformation of it, an affine invariant fitting algorithm generates two fittings to the two data sets such that the resulting implicit polynomials are also related by the same affine transformation.

As mentioned in section 2, the 3L fitting algorithm is inherently Euclidean but not affine invariant. There are two possible cures to this problem: removing the "affineness' of the data set by a scatter matrix normalization or replacing the distance in the level set generation by an affine invariant quantity.

3.1 Data Set Normalization

The scatter matrix (sample covariance matrix) Σ of a data set is a symmetric positive matrix. It can be rewritten as $\Sigma = Q\Delta Q^T$ where Q is an orthogonal matrix of normalized eigenvectors of Σ and Δ is the diagonal matrix of corresponding eigenvalues. If the transformation $A_w = \Delta^{-1/2}Q^T$ is applied to the data points the scatter matrix of the resulting data set becomes the identity matrix I. The application of transformation A_w , sometimes called *whitening*, changes the dispersion of the data by making the spectrum of eigenvectors uniform.

If Γ_0 and $\hat{\Gamma}_0$ are two data sets related by an affine transformation, applying a whitening on both reduces the mathematical transformation between them to a rotation. Therefore, after the whitening the "affiness" is removed and 3L fitting can be used without needing any modification.

An obvious shortcoming of the data set normalization is that subpixel pixel quantization, occlusion and other natural effects inherent in the image acquisition makes the scatter matrix unreliable. In addition, the assumption that the two data sets are pointwise in 1-1 correspondence is not realistic because of sampling and quantization in the formation of 2D images.

3.2 Affine Invariant Level Set Generation

In the affine case the primary problem with the 3L fitting is the use of Euclidean distance a non-affine quantity, in generating level sets from a given data set. Our remedy to this problem is to replace the distance by an *affine invariant quantity*. We propose two such quantities: *the line-segment ratio*, and *the affine curvature*.

3.2.1 Line Segment Ratio

Under an affine transformation, lines become lines. Although the lengths of individual line-segments are not preserved, the ratio of lengths of any two line-segments remains the same. This property can be used to formulate an affine invariant level set generation technique. As illustrated in Figure 1, from a given covariant point

P_c, a line segment $| P_c P_i |$ (with length l) is drawn to a point P_i is on the curve. Line-segments, such as $| P_i P_{-i} |$ and $| P_i P_{+i} |$ in the figure, along the same direction and starting at P_i with lengths c a constant multiple k of l (the length of $| P_c P_i |$) and $|P_c P_i|$ form a pair for which line-segments ratio is an affine invariant. Therefore for a given c, the end points of the two line-segments starting at P_i to either side of the curve give two points P_{+i} and P_{-i} on the two level sets. The level sets obtained this way will be affine invariant. The value k can be chosen as a certain constant ratio (such as 5%) of the contour point-covariant point distance l. Here the covariant point P_c can be the data mean or the intrinsic coordinate center obtained by fitting a preliminary implicit curve or another point with known coordinates before and after the affine transformation.

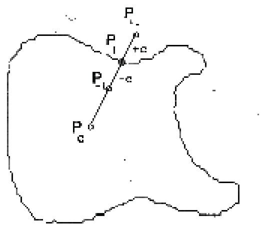

Figure 1: Line-segment ratio level set generation

A drawback of this approach is its dependencies on the robustness of the covariant point P_c. If, for instance, the data mean is chosen to be the reference covariant point it is sensitive to arbitrary occlusions of the data set. Instead, an intrinsic coordinate center, obtained by fitting a preliminary implicit curve, can be used through an iterative approach. That is, first, a preliminary implicit curve is fitted to the data set and the resulting representation can be used to obtain the intrinsic object center which will then be used as the covariant point for computing an affine invariant fit.

3.2.2 Affine Curvature

Recently, the planar curve evolution, based on variants of the classical heat flow equation, was used for shape representation, smoothing and decomposition. Fundamental to the curve evolution is invariant flows with respect to certain transformations. According to this approach, in affine invariant flows a point on the curve changes position in the direction normal to the curve and with a velocity proportional to the affine curvature at the point. Analogous to the Euclidean curvature κ, a Euclidean differential invariant, the affine curvature can be approximated by $\kappa^{1/3}$ where κ is the Euclidean curvature.

Given y=u(x), an explicit function representing the data curve $\Gamma_0 = \{(x_i, y_i) \mid i = 1,...K\}$, below in Table 1 are the curvatures for some transformation groups

Group	Curvature
Euclidean	$\dfrac{u_{xx}}{(1+u_x^2)^{3/2}}$
Similarity	$\dfrac{(1+u_x^2)u_{xxx} - 3u_x u_{xx}^2}{u_{xx}^2}$
Special Affine	$\dfrac{P_1}{u_{xx}^{8/3}}$
Affine	$\dfrac{P_2}{P_1^{3/2}}$

Table 1. Curvatures for some transformation groups

where $P_1 = 3u_{xx}u_{xxxx} - 5u^2_{xxx}$,

$P_2 = 9u^2_{xx}u_{xxxxx} - 45u_{xx}u_{xxx}u_{xxxx} + 40u^3_{xxx}$

A *special affine transformation* is an affine transformation for which the determinant of the 2 x 2 matrix A is equal to unity. Note that since the determinant is unity, all invariants of the group of special affine transformations (without translations) are absolute weight invariants, as is the case for the group of rotations.

These ideas have a direct implication for affine-invariant level set generation. With this approach, for a given curve point P_i having affine curvature τ_i and normal $\vec{N}_i$, two points each τ_i away from P_i along and opposite to the normal direction to the curve, form the corresponding inside and outside level set points. The level set generated so will be affine invariant. The affine curvature τ_i can also be used in other ways, as a constant in determining level value (the distance) at a point.

Determining the affine curvature, directly or as an approximation from the Euclidean curvature, is somewhat involved because noise, perturbations and occlusion make the local curvature computation, based on difference equations unstable. In medical imaging, this problem is circumvented by first computing a Euclidean distance transform of the curve in the vicinity of the curve, i.e., finding the distance of each pixel (within a given neighborhood) from the curve, and then using this information for the curvature estimation. Analogous to the level set generation in the 3L fitting, the distance transform around the curve forms an explicit surface which makes the curvature estimation more stable. With the explicit surface x = f(x,y), the Euclidean curvature is defined as

$$\kappa_1 = -\frac{f_{xx}f_y^2 - 2f_{xy}f_x f_y + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{3/2}}$$

The main disadvantage of this technique is the computational cost of computing the distance transform. Another alternative is fitting an explicit or implicit polynomial locally or globally and using it to determine the curvature at a given point.

4. Experiments

In this section, we compare the performance of the Euclidean level set generation (distance transform) with the performance of our line-segment-ratio level set generation technique under affine transformations. For this purpose we used the shapes shown in Figure 2.

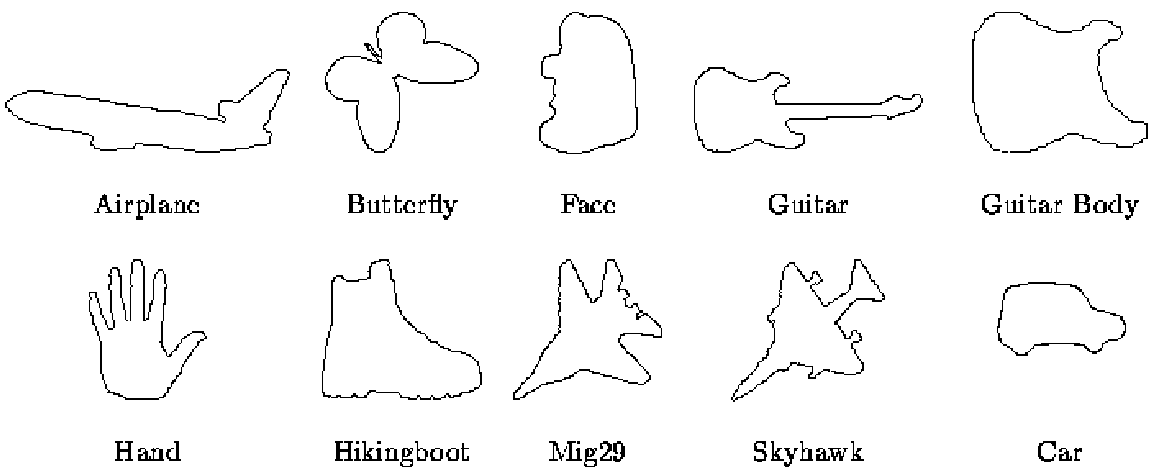

Figure 2: Objects used in the experiments.

For each shape, first a 4^{th} degree implicit polynomial is fitted to the given pixel level data set. Then, 100 random affine transformations are generated and the data set is transformed with these affine transformations. For each transformation, the resulting data set is quantized from subpixel values to pixel values by rounding the x and y coordinates of the transformed points. This introduces a quantization noise. Then, a 4^{th} degree implicit curve is fitted to the data set obtained in this way. Ideally, the polynomial obtained at this stage should be related to the polynomial representation for the original data set by the same affine transformation. That is, two coefficient vectors should be the same up to the given affine transformation. We transform the coefficient vector, for the polynomial fit to transformed data, back to its normal position and try to determine the distribution for each coefficient (100 samples from the 100 affine transformations) by computing means and standard deviations. Theoretically the distributions should be an impulse with zero variance; therefore, smaller variance indicate better results. The experiment is run with each of the two level set generation techniques – which are applied prior to the implicit polynomial fittings – and their performances are compared. Table 2 gives the actual values (the value for the original data set), means and standard deviations of the individual coefficients for the airplane object with the two level set generation techniques. Similar results for other datasets are given in [6]. Note that the corresponding coefficients may have different values (the same rows of the coefficient columns in the tables (a) and (b) of each shape) with the two level set generations because, although the data set Γ_0 is the same Γ_{+c} and Γ_{-c} can be quite different with different level set generation techniques. The results indicate that the level set generation with the line-segment ratio technique gives much less coefficient variability.

Coefficient	Actual	Mean	Std
a_{00}	-1.0202	-1,0623	0.3221
a_{10}	2.8073	2.7016	0.3519
a_{01}	43.8368	42.1145	2.2066
a_{20}	-30.7494	-29.3235	5.6359
a_{11}	51.4314	48.1990	9.2220
a_{02}	202.1709	204.0876	19.5688
a_{30}	5.0325	4.6849	1.0972
a_{21}	-192.3056	-184.6405	19.6451
a_{12}	-484.5704	-455.7016	87.2006
a_{03}	602.6677	570.0975	281.3221
a_{40}	61.2979	58.4404	10.0538
a_{31}	65.9884	61.0090	20.6475
a_{22}	21.6180	14.4137	43.4018
a_{13}	-2912.5210	-2682.0589	853.7351
a_{04}	6890.2700	6224.4336	1603.6172

Coefficient	Actual	Mean	Std
a_{00}	0.0625	0,1509	0.0641
a_{10}	2.1189	1.5735	0.4789
a_{01}	31.2716	29.6464	1.0376
a_{20}	-32.5868	-31.1479	2.0945
a_{11}	49.5480	50.4607	3.7091
a_{02}	88.1154	65.9655	17.2355
a_{30}	6.2323	6.8007	0.5549
a_{21}	-162.5114	-151.1471	10.1599
a_{12}	-477.2975	-436.1843	39.6397
a_{03}	692.4871	615.2459	118.4975
a_{40}	61.4914	58.6781	3.6577
a_{31}	84.3208	71.3743	10.2151
a_{22}	258.6494	254.1725	20.5205
a_{13}	-3441.6466	-3316.8648	265.9839
a_{04}	7278.0686	7165.4870	404.7065

Table 2. Airplane coefficients with (a) Euclidean levels (b) Line-segment ratio levels

5. Summary and Conclusions

The line-segment ratio based level set generation technique is easily implementable. Yet the stability of the results depends on the robustness of the covariant point. The drawback of the affine curvature technique, however, is the difficulty of computing a stable affine curvature, directly or through the Euclidean curvature approximation which is only approximately affine invariant. As mentioned above, the curvature estimation through distance transform formation is computationally costly. If, instead, local explicit or implicit polynomials are used to determine the curvature, the curvature may not be smooth (a desired property for the level sets) especially when higher degree polynomials are used.

Nevertheless, the experimental results indicate that one of the proposed techniques improves considerably on using the Euclidean distance transform for the level set generation and the other technique looks promising. More experiments are needed to be run, including a test of the affine curvature based level set generation technique, and testing the performance of both techniques under occlusion.

REFERENCES

[1] Civi, H., "Implicit Algebraic Curves and Surfaces for Shape Modelling and Recognition," Ph.D. Dissertation, Bogazici University, October 1997.

[2] Keren, D., D. Cooper, and J. Subrahmonia, "Describing complicated objects by implicit polynomials," IEEE PAMI, pp. 38-53, January 1994.

[3] Lei, Z., M. M. Blane and D. B. Cooper. "3L Fitting of Higher Degree Implicit Polynomials," In Proceedings of Third IEEE Workshop on Applications of Computer Vision, Sarasota, FL, December 1996.

[4] Tasdizen, T., J. P. Tarel and D. B. Cooper, "Algebraic Curves That Work Better", LEMS Tech. Report 176, Division of Engineering, Brown University, 1998. IEEE Transactions on Image Processing.

[5] Unsalan, C., and A. Ercil. "A new robust and fast implicit polynomial fitting technique" Research paper FBE-IE-15/98-19, Bogazici University, November 1998.

[6] Yalniz, C., and A. Ercil. "Implicit Polynomials for Invariant Object Recognition" Research paper FBE-IE-15/00-06, Bogazici University, April 2000.