Pattern Recognition Letters 14 (1993) 799–808
North-Holland

October 1993

PATREC 1089

A note on the least squares fitting of ellipses

Paul L. Rosin

Cognitive Systems Group, School of Computing Science, Curtin University of Technology, Perth, 6001, Western Australia

Received 4 February 1992

Abstract

Rosin, P.L., A note on the least squares fitting of ellipses, Pattern Recognition Letters 14 (1993) 799–808.

The characteristics of two normalisations for the general conic equation are investigated for use in least squares fitting: either setting $F=1$ or $A+C=1$. The normalisations vary in three main areas: curvature bias, singularities, transformational invariance. It is shown that setting $F=1$ is the more appropriate for ellipse fitting since it is less heavily curvature biased. Setting $A+C=1$ produces more eccentric conics, resulting either in over-elongated ellipses or hyperbolae. Although the $F=1$ normalisation is less well suited than the $A+C=1$ normalisation with respect to singularities and transformational invariance both these problems are solved by normalising the data, shifting it so that it is centred on the origin before fitting, and then re-expressing the fit in the original frame of reference.

Keywords. Ellipse, least squares fitting, normalisation, curvature bias, singularity, transformational invariance.

Introduction

An important part of early visual processing is the extraction of features from the image. A common feature are curves, derived from edges. To be more easily manipulated by the following stages of reasoning they need to be represented by higher-order primitives than the individual pixels. One such representation that is useful for computer vision is the ellipse. If ellipses detected in the image are assumed to be the 2D projection of circular features in the 3D scene then various inferences can be made about their 3D structure. These restrict the number of interpretations of the data, and can be used for matching [28], grouping [21], and cueing [19].

Compared to simple representations such as straight lines, ellipses and other higher-order representations contain many more parameters. This makes segmenting the curves and fitting the representation more complex. We have discussed techniques for segmenting curves into ellipses in previous papers [20,27]. Here we consider the fitting of the ellipses to the segmented data. There are two main methods for fitting ellipses. (1) Accumulating evidence followed by searching for peaks in the accumulator space (i.e., the Hough Transform [11,25]). (2) Fitting by minimising some error function. (An interesting combination of the two approaches is given by [26].) The Hough Transform is not well suited to analysing edge lists for several reasons: it does not produce a unique solution; the search space is multi-dimensional for higher-order representations, which increases and complicates computation; and it requires a large number of data points for good results. This last consideration is important since

Correspondence to: Dr. Paul Rosin, Terminale Postale 670, Joint Research Centre, 2120 Ispra, Varese, Italy.

we may wish to fit ellipses to a small number of significant data points on the curve to increase efficiency [20].

If the data is already segmented then error minimisation fitting techniques are appropriate. However, this involves several problems if the best ellipse is to be fit to any data since an ellipse is described by the general equation for a conic

$$Ax^2+Bxy+Cy^2+Dx+Ey+F=0$$

with the constraint $B^2-4AC<0$. The problem of constraining the fitted conic to be elliptical is non-linear, requiring iterative solutions. The main options when fitting conics are to:

(1) allow all conic types (hyperbolic, parabolic and elliptic) [2,5,12,15];

(2) ignore non-elliptic fits [9,10,17,18,22];

(3) force the conic fit to be an elliptic arc [4,7,8,20];

(4) reject all non-elliptic conic fits, typically replacing them by a simpler representation [13,14,24,27].

For many applications in computer vision option (1) is not appropriate since hyperbolae and parabolae are not useful features. Option (2) is not suitable either since arbitrary curves are likely to generate many non-elliptic fits, and must still be represented somehow. Our earlier algorithm for segmenting curves into combinations of lines and ellipses forced the conic fit to be an ellipse [20]. However, an iterative technique is required which is computationally expensive. It was noted that most ellipses fitted to non-elliptic data were discarded since the data is better fit by some other representation (e.g. straight lines), or by several separate ellipses. Therefore, in the interests of efficiency it may be preferable to use a simple and efficient conic fitting technique at the cost of failing to fit elliptic conics to some data since these will be better fitted in other ways in any case [27]. The efficiency of the fitting technique is important to applications such as curve segmentation since they require fits to be made to many different sections of the data and the total computation time of the algorithm is dominated by the fitting time.

This paper describes the least squares fitting of ellipses. First, the problems inherent in the fitting are introduced. Then two common normalisations of the conic equation are discussed, their characteristics are compared, and the most suitable for ellipse fitting is determined.

Least squares ellipse fitting

The general equation of a conic is

$$Q(x,y)=Ax^2+Bxy+Cy^2+Dx+Ey+F=0$$

The conic which best fits a set of n points can be determined using the least squares error criteria which minimises the pointwise error

$$S=\sum_{i=1}^{n} d_i^2$$

where d_i is the distance of a point to the conic. This is problematic since the *Euclidean* distance d_i is difficult to evaluate. A common approximation is the *algebraic* distance $Q(x_i, y_i)$, and so the function to minimise becomes

$$S=\sum_{i=1}^{n} Q(x_i,y_i)^2 .$$

This introduces a second problem: the need to avoid the trivial solution

$$A=B=C=D=E=F=0 .$$

One method is to use the 'average gradient constraint' [1]. This requires that the mean square gradient of $Q(x_i, y_i)$ measured at all points sums to one. However, the minimisation that follows requires complex numerical techniques. A computationally simpler method is to normalise $Q(x,y)$. Many different normalisations have been used for conic fitting [6]. Since a simple efficient implementation is required only linear normalisations will be considered. The most common normalisations are to set $F=1$ (e.g. [2,8,27]) or to set $A+C=1$ (e.g. [17,20]). Setting $F=1$ gives

$$Q(x,y)=Ax^2+Bxy+Cy^2+Dx+Ey+1=0 .$$

Obtaining the partial derivatives with respect to A, B, C, D and E, and setting these to zero

$$\frac{\partial S}{\partial A}=\sum_{i=1}^{n} 2x_i^2(Ax_i^2+Bx_iy_i+Cy_i^2+Dx_i+Ey_i+1)=0 ,$$

$$\frac{\partial S}{\partial B} = \sum_{i=1}^{n} 2x_i y_i (Ax_i^2 + Bx_i y_i + Cy_i^2 + Dx_i + Ey_i + 1) = 0 ,$$

etc.

provides five simultaneous equations, the solution to which represents the conic that minimises the error function S:

$$A \sum x^4 + B \sum x^3y + C \sum x^2y^2 + D \sum x^3 + E \sum x^2y + \sum x^2 = 0 ,$$

$$A \sum x^3y + B \sum x^2y^2 + C \sum xy^3 + D \sum x^2y + E \sum xy^2 + \sum xy = 0 ,$$

$$A \sum x^2y^2 + B \sum xy^3 + C \sum y^4 + D \sum xy^2 + E \sum y^3 + \sum y^2 = 0 ,$$

$$A \sum x^3 + B \sum x^2y + C \sum xy^2 + D \sum x^2 + E \sum xy + \sum x = 0 ,$$

$$A \sum x^2y + B \sum xy^2 + C \sum y^3 + D \sum xy + E \sum y^2 + \sum y = 0 .$$

Alternatively, setting $A+C=1$ gives

$$Q(x, y) = Ax^2 + Bxy + (1-A)y^2 + Dx + Ey + F = 0$$

which can be rewritten as

$$Q(x, y) = A(x^2 - y^2) + Bxy + Dx + Ey + F + y^2 = 0 .$$

The five simultaneous equations to be solved are obtained as before:

$$A \sum (x^2-y^2)^2 + B \sum (x^2-y^2)xy + D \sum (x^2-y^2)x + E \sum (x^2-y^2)y + F \sum (x^2-y^2) + \sum (x^2-y^2)y^2 = 0 ,$$

$$A \sum xy(x^2-y^2) + B \sum x^2y^2 + D \sum x^2y + E \sum xy^2 + F \sum xy + \sum xy^3 = 0 ,$$

$$A \sum x(x^2-y^2) + B \sum x^2y + D \sum x^2 + E \sum xy + F \sum x + \sum xy^2 = 0 ,$$

$$A \sum y(x^2-y^2) + B \sum xy^2 + D \sum xy + E \sum y^2 + F \sum y + \sum y^3 = 0 ,$$

$$A \sum (x^2-y^2) + B \sum xy + D \sum x + E \sum y + Fn + \sum y^2 = 0 .$$

Comparing normalisations

The effects of the two normalisations $F=1$ and $A+C=1$ on the performance of the ellipse fitting will be compared in this section. Care has to be taken in choosing the normalisation since most normalisations create singularities. These prevent a set of conics being fit, and make the neighbouring conics difficult to fit. Setting $F=1$ has as singularities all conics that go through the origin. Setting $A+C=1$ has no singularities involving ellipses – only rectangular hyperbolae are excluded. This implies that the latter normalisation is the more appropriate one for ellipse fitting. However, changing $Q(x, y)$ may also change the resulting fits even if the conic is far from the singularities since the whole quadratic surface that describes the error is altered. This makes it hard to gain any intuitive feel for the effects of the normalisation on the conic fit. We shall examine the results of the effects of the two normalisations on several examples of fitting conics to data and compare the normalisations based on this empirical evidence.

It was stated earlier that the algebraic distance $Q(x, y)$ could be used to approximate the Euclidean distance. Bookstein [5] showed that

$$Q(x_i, y_i) \propto \frac{p_i^2}{c_i^2} - 1$$

where p_i is the distance from the point (x_i, y_i) to the centre of the conic, and c_i is the distance from the conic to its centre along the ray from centre to point. Thus this measure retains a qualitative similarity to the Euclidean distance since it increases monotonically with distance. However, its rate of increase is greater near low curvature sections of the ellipse than at higher curvature sections. Figure 1 illustrates this by showing error contours (alternatively coloured black and white) drawn about an ellipse (the thin gray contour). The change in the gradient of $Q(x, y)$ causes data points closer to the low curvature sec-

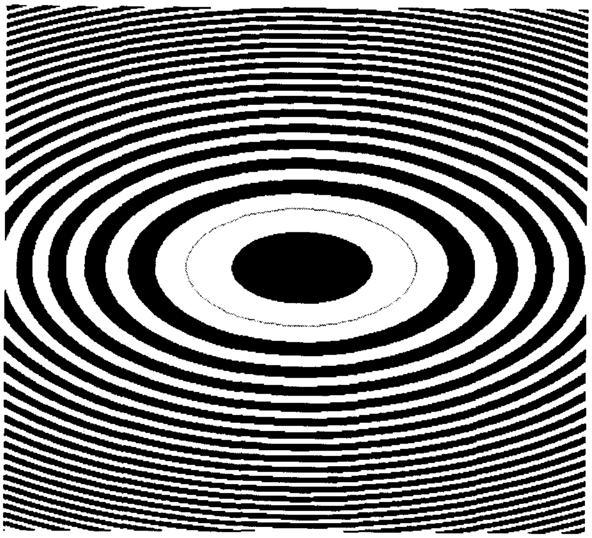

Figure 1. Distance measure $Q(x_i, y_i)$ showing bars of similar value from the ellipse (the thin grey contour).

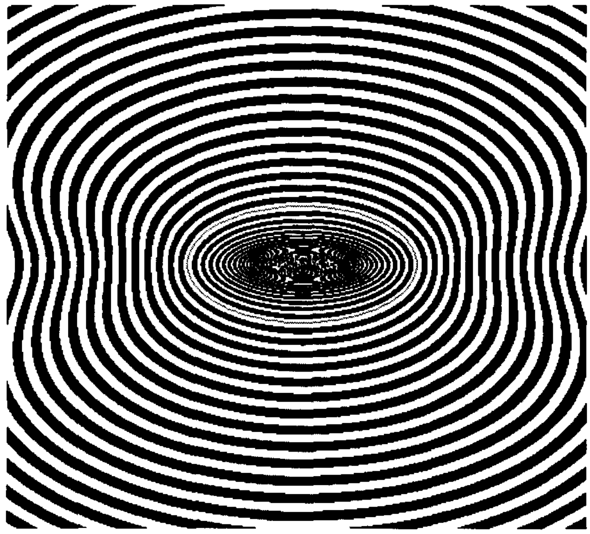

Figure 2. Distance measure $Q(x_i, y_i)/|\nabla Q(x_i, y_i)|$.

tions to have a greater effect on the resultant fit than other points. This bias causes the eccentricity of the fitted ellipse to be overestimated. The ambiguous term 'high curvature bias' (e.g. [3]) is often used to describe the bias.

Better approximations exist to the Euclidean distance. The most common is obtained by inversely weighting $Q(x, y)$ by its gradient [1]

$$d_i \approx \frac{Q(x_i, y_i)}{|\nabla Q(x_i, y_i)|}.$$

This distance measure is plotted as before in Figure 2. The error contours are approximately uniformly wide although they become distorted further away from the ellipse. As the ellipse becomes more eccentric the inaccuracy of the distance approximation increases. The error contours in Figure 3 show significant distortion near the pointed ends of the ellipse.

Another distance measure is obtained by weighting $Q(x_i, y_i)$ by the distance c_i from the conic to its centre along the ray from centre to point (x_i, y_i). The higher the curvature the greater this distance, and so the curvature bias is offset to some degree. In fact, the curvature bias is over-corrected and it can be seen from the error contours in Figure 4 that it gives data points closer to the high curvature sections a greater weight, causing the eccentricity of the fitted ellipse to be under-estimated. In Ballard&Brown's terminology this is a 'low curvature bias'. Although the bias is greater than the previous measure it does not have the extreme distortions for elongated ellipses as shown in Figure 5. In addition, the gradient of this distance is more uniform than that of the algebraic distance.

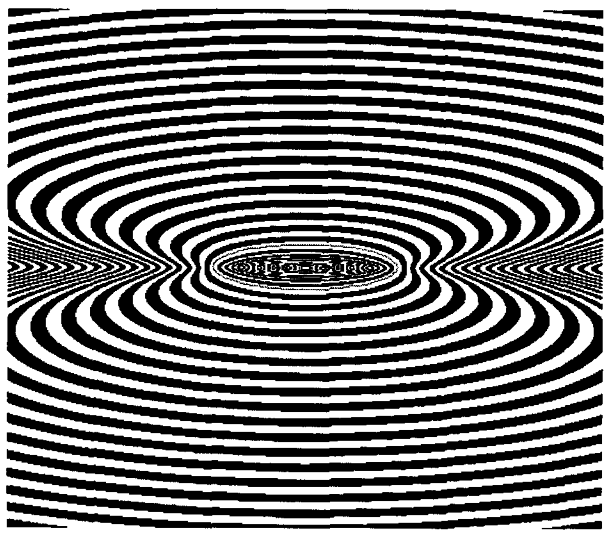

Figure 3. Distance measure $Q(x_i, y_i)/|\nabla Q(x_i, y_i)|$ on elongated ellipse.

An alternative to directly improving the distance measure is to use the simple algebraic distance $Q(x_i, y_i)$ and iteratively reweight each data point by

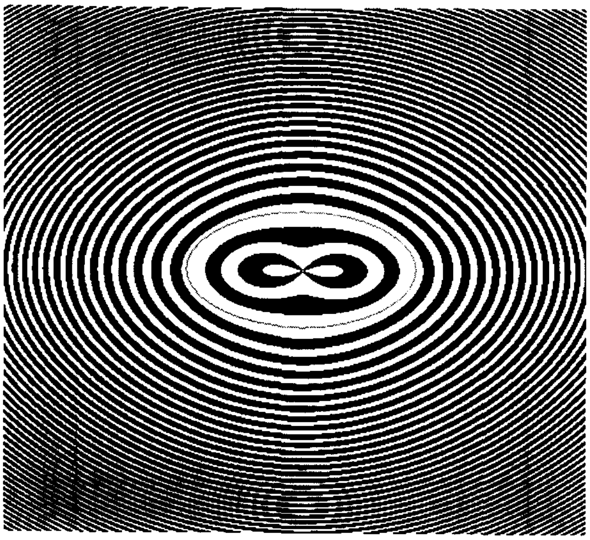

Figure 4. Distance measure $c_i Q(x_i, y_i)$.

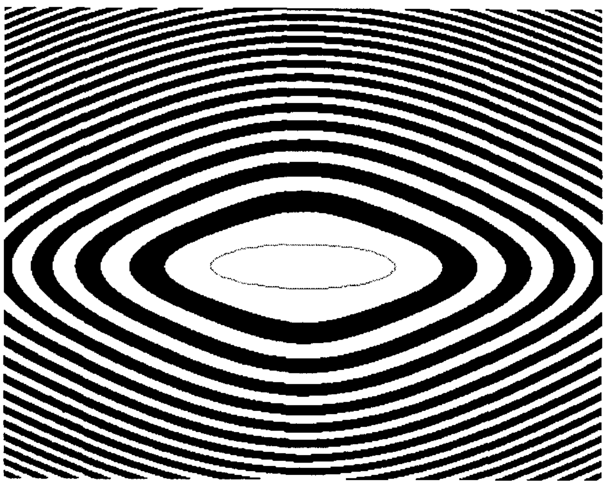

Figure 5. Distance measure $c_i Q(x_i, y_i)$ on elongated ellipse.

the weights above; i.e., the inverse gradient of the conic equation $1/\nabla Q(x_i, y_i)$ [23] or the distance from the conic centre to the conic c_i along the ray to the point (x_i, y_i) [22].

The disadvantage both of the alternative distance measures and reweighting the data points is that iterative methods are required to minimise the error which reduces computational efficiency. Therefore we will use the algebraic distance measure at the cost of its curvature bias to retain efficiency since it has a closed-form solution. However, two factors can reduce this bias: the choice of normalisation; and the selection of data points. The following examples compare the resulting fits obtained when the two normalisations are used. In all the following figures the fitted ellipses resulting from the $F=1$ normalisation are plotted in black and the fitted ellipses from the $A+C=1$ normalisation are plotted in grey. Figure 6 shows noisy data artificially generated for short sections of four ellipses with increasing eccentricity. Short sections are used since curvature bias is most apparent on sections of data making up less than half the total ellipse. It can be seen that the $A+C=1$ normalisation consistently produces more eccentric ellipses than the $F=1$ normalisation.

Noisy data from sections of an ellipse with high eccentricity are shown in Figures 7–9 with conic fits using both normalisations. As the sections of the ellipse are shortened both the ellipse fits tend to become more eccentric due to the curvature bias. Eventually, using the $A+C=1$ normalisation the shortest section (Figure 9) has been fit by a hyperbola. This is because of the bias to overestimating the eccentricity of the fitted conic. The eccentricity of an ellipse lies between 0 and 1, and the eccentricity of a hyperbola is greater than 1. Thus increased eccentricity implies a tendency to fit hyperbolae in place of ellipses.

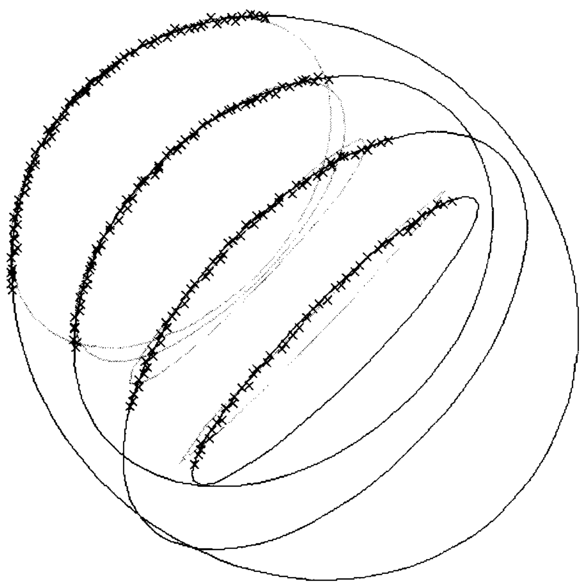

Figure 6. Noisy data for sections of 4 ellipses with fits using $F=1$ and $A+C=1$ normalisations.

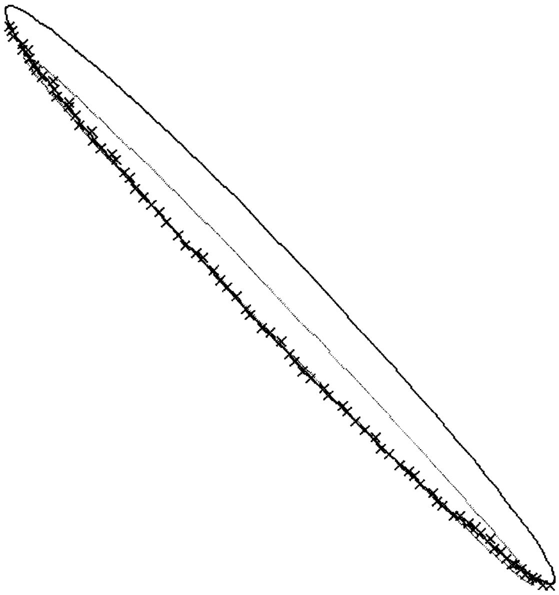

Figure 7. Noisy data for a section of an elongated ellipse with fits using both normalisations.

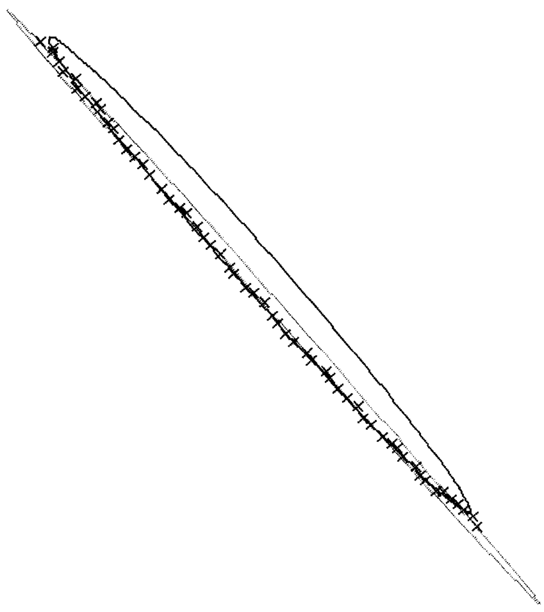

Figure 8. Noisy data for a section of an elongated ellipse with fits using both normalisations.

It is important to note that the selection of data points can alter the fit. The curve segmentation algorithm described in [20,27] first fits straight lines to the pixel data and then fits ellipses to the endpoints of the lines. Fitting ellipses to endpoints rather

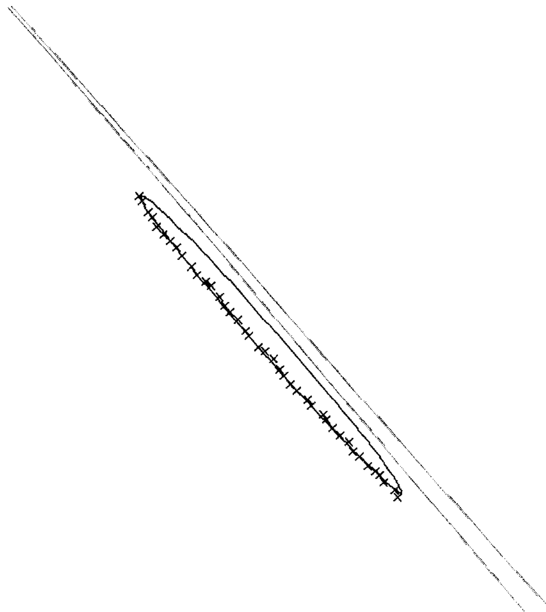

Figure 9. Noise data for a section of an elongated ellipse with fits using both normalisations.

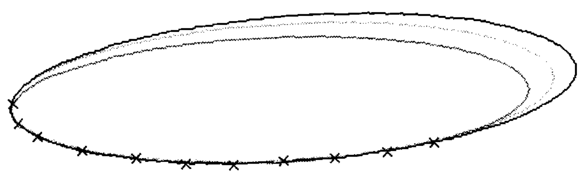

Figure 10. Regularly sampled data from a section of an ellipse with both fits.

than pixels increases efficiency since a much smaller amount of data is used. The line fitting stage has the property of fitting shorter lines to sections of high curvature data than to low curvature sections. This results in a sampling of data points approximately proportional to the curvature. We have already noted that the algebraic distance measure causes points at high curvature to be given a lower weighting. The increased data sampling at these areas increases the weighting, and will therefore offset the curvature bias in the fitting to some degree. This is shown in Figures 10 and 11 in which the original ellipse (plotted in light grey) is used to generate a list of pixels with added noise. In Figure 10 the data was sampled at regular intervals. In Figure 11 the data was first approximated by straight lines [20] and the endpoints

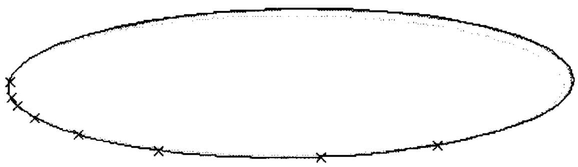

Figure 11. Data derived from endpoints of lines with both fits.

were used for ellipse fitting. The fit to the line endpoints is more accurate, showing less bias to over-elongation even though less data was used. However, it should be noted that with such small numbers of data points the fit is unstable. Although the data is consistently well fitted, the extrapolated ellipse is very sensitive to small changes in the data values. This is particularly evident for short sections of elliptic data [17]. This suggests that intermediate fits during segmentation may be satisfactorily performed on line data, but that the final retained fits should be refined using the additional pixel data.

A significant problem for the $F=1$ normalisation is its singularity preventing accurate conic fits to data through the origin since this requires setting $F=0$. In fact, when setting $F=1$ good fits can be made to noise-free data through the origin since it is the *ratios* of the values of the coefficients that are important. An example is shown in Figure 12. Although F is fixed at 1, the other coefficients have become large, making F proportionally small. The values of the coefficients are (-1243.30, 2227.46, -1286.90, 31.88, -324.88, 1.00) as compared to the fit using the $A+C=1$ normalisation whose coefficients are (0.49, -0.88, 0.51, -0.13, 0.13, 0.00). Normalising the first set of coefficients so that $B=-0.88$ produces an identical set (to two decimal places) of coefficients to those from the second fit. However, when noise is added to the data, problems arise. Figure 13 shows that the fit with the $F=1$ normalisation has avoided the origin. As the data becomes noisier the fit becomes significantly worse (Figure 14). A simple solution to this problem is to normalise the data by shifting it so that the centroid of the data lies on the origin. After the conic fitting it is shifted back to the original frame of reference of the data. If the coefficients of the conic are reexpressed as the centre, major and minor axis lengths, and orientation of the major axis of the ellipse, the second shift is straightforward. This technique is often carried out as standard practise prior to fitting to reduce numerical errors. When this technique is used the $F=1$ normalisation gives an accurate fit even for the very noisy data as shown in Figure 15.

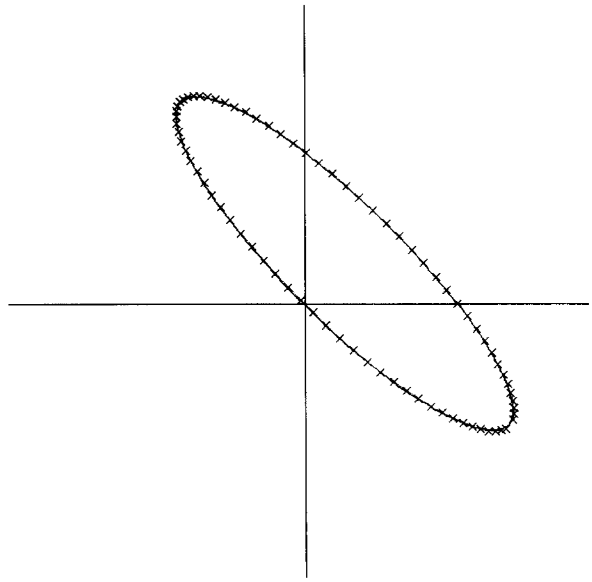

Figure 12. Both fits to noise-free data through origin.

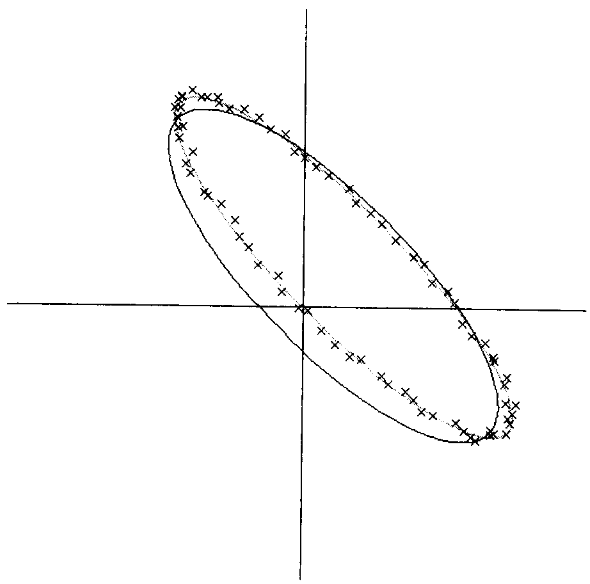

Figure 13. Both fits to noisy data through origin.

Another difference between the normalisations is their response to the transformation of the data. The

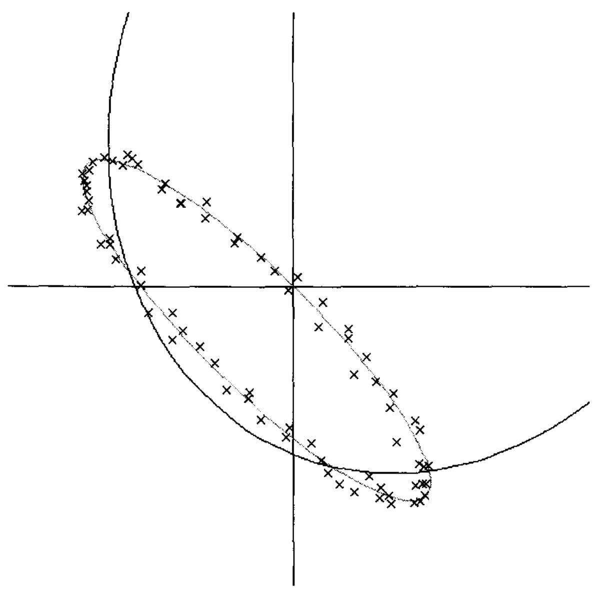

Figure 14. Both fits to very noisy data through origin.

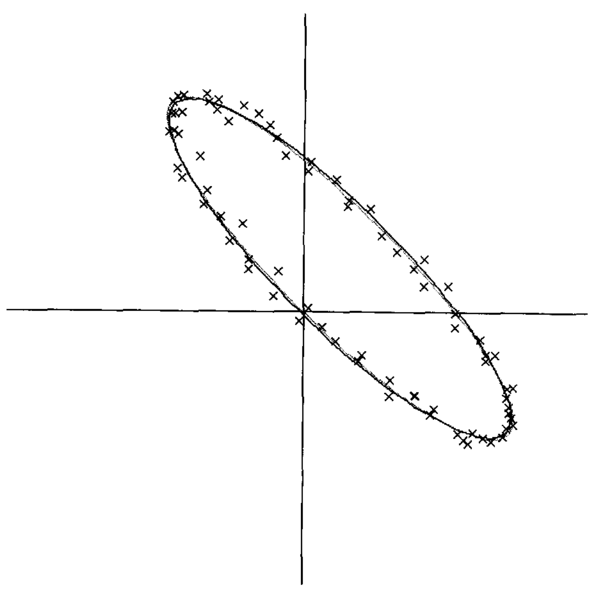

Figure 15. Both fits to noisy data using data normalisation.

$A+C=1$ normalisation is invariant to translations and rotations whereas the $F=1$ normalisation is not. Transformational invariance of ellipse fitting is a desirable property since it ensures that the following stages of processing that use the ellipse (e.g. matching, curve segmentation) are invariant to transformations of the data. This indicates that the $A+C=1$ normalisation is superior to the $F=1$ normalisation in this respect. However, the $F=1$ normalisation is made translation invariant by normalising the data as above, i.e., shifting it to the origin before ellipse fitting. Fitting is still not rotation invariant but experimental results indicate that any variations are minor. For instance, Figure 16 shows data for a noisy ellipse rotated by different amounts with their fits which are almost identical. The parameters of the fits vary by less than 1%.

Conclusions

Since ellipse fitting is a non-linear problem it is often simplified to the fitting of conics. In this case there exists for least squares fitting closed-form rather than iterative solutions. These have the advantages of simplicity of implementation and computational efficiency. Conic fitting is acceptable in many applications which require ellipse if the occasional non-elliptic conic fit to the data can be tolerated.

Two problems arise with least squares fitting. (1) An approximation to the Euclidean distance must be taken – the algebraic distance is the most convenient one but has a bias to weight low curvature points more than high curvature points. (2) To prevent trivial so-

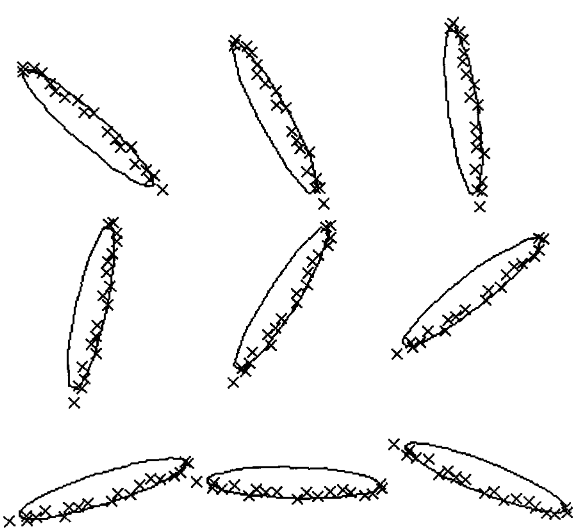

Figure 16. Fits to rotated data (F set to 1).

lutions the general conic equation must be normalised, usually introducing singularities which distort certain fits. The characteristics of two normalisations are compared with respect to curvature bias, singularities, and transformational invariance.

Both normalisations are biased to weight points at low curvature sections more than high curvature points, causing the eccentricity of the ellipse to be over-estimated. The bias to over-estimating eccentricity results in a greater tendency to fit over-elongated ellipses or to fit hyperbolae in place of ellipses. Since hyperbolic fits may be made to elliptical data due to the presence of noise this bias is undesirable. It is shown that setting $F=1$ is more appropriate for ellipse fitting than setting $A+C=1$ since it produces a smaller over-estimation of conic eccentricity.

At first sight the $A+C=1$ normalisation appears superior to the $F=1$ normalisation with respect to singularities and transformational invariance. The singularity associated with the $F=1$ normalisation often prevents accurate fits to conics that pass through the origin. However, this can usually be overcome by shifting the data so that it is centred on the origin before fitting, and then re-expressing the fit in the original frame of reference. This procedure is often carried out anyway as standard practise to reduce numerical errors. At the same time, normalising the data provides invariance to translations of the data. It is shown that rotations of the data have little effect in the resulting fits.

Acknowledgements

Many thanks to Geoff West, Ian Wright, and Steve Carnie for many helpful discussions.

References

[1] Agin, G.J. (1981). Fitting ellipses and general second-order curves. Tech. Report CMU-RI-TR-81-5, Robotics Institute, Carnegie-Mellon Univ.

[2] Albano, A. (1974). Representation of digitised contours in terms of conic arcs and straight-line segments. *Computer Vision, Graphics, and Image Processing* 3, 23–33.

[3] Ballard, D.H. and C.M. Brown (1982). *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ.

[4] Bolles, R.C. and M.A. Fischler (1981). A RANSAC-based approach to model fitting and its applications to finding cylinders in range data. *Proc. 7th IJCAI*, 637–643.

[5] Bookstein, F.L. (1979). Fitting conic sections to scattered data. *Computer Vision, Graphics, and Image Processing* 9, 56–71.

[6] Carnie, S.L. (1990). A survey of least square ellipse-fitting algorithms. Computing Project 594 Report, Grad. Dip. Computing, School of Computing Science, Curtin Univ. of Technology.

[7] Cooper, D.B. and N. Yalabik (1975). On the cost of approximating and recognizing a noise perturbed straight line or a quadratic curve segment in the plane. Tech. Report NASA NSG5036/1, Brown Univ.

[8] Cooper, D.B. and N. Yalabik (1976). On the computational cost of approximating and recognizing noise-perturbed straight lines and quadratic arcs in the plane. *IEEE Trans. Comp.* 25, 1020–1032.

[9] Forsyth, D., J.L. Mundy, A. Zisserman and C.M. Brown (1990). Projectively invariant representations using implicit algebraic curves. In: O. Faugeras, Ed., *Computer Vision – ECCV 90.* Springer, Berlin, 427–436.

[10] Foster, N.J. and A.C. Sanderson (1984). Determining object orientation using ellipse Fitting. *SPIE Intelligent Robots and Computer Vision* 521, 34–43.

[11] Huang, C.L. (1989). Elliptical feature extraction via an improved Hough transform. *Pattern Recognition Lett.* 10, 93–100.

[12] Liao, Y-Z. (1981). A two-stage method of fitting conic arcs and straight-line segments to digitized contours. *Proc. Pattern Recognition and Image Processing Conf.*, 224–229.

[13] Nagata, T., H. Tamura and K. Ishibashi (1985). Detection of an ellipse by use of a recursive least-squares estimator. *J. Robotic Systems* 2, 163–177.

[14] Nakagawa, Y. and A. Rosenfeld (1979). A note on polygonal and elliptical approximation of mechanical parts, *Pattern Recognition* 11, 133–142.

[15] Nalwa, V.S. and E. Pauchon (1987). Edge aggregation and edge description. *Computer Vision, Graphics, and Image Processing* 40, 79–94.

[16] Pavlidis, T. (1983). Curve fitting with conic splines. *ACM Trans. Graphics* 2, 1–31.

[17] Porrill, J. (1990). Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter. *Image and Vision Computing* 8, 37–41.

[18] Richetin, M., J.T. Lapreste and M. Dhome (1985). Recognition of conics in contours using their geometric properties. *Proc. CVPR*, 464–469.

[19] Rosin, P.L. (1993). Acquiring information from cues. *Pattern Recognition Lett.* 14, 599–609.

[20] Rosin, P.L. and G.A.W. West (1990). Segmenting curves into elliptic arcs and straight lines. *IEEE Internat. Conf. on Computer Vision*, Osaka, Japan, 75–78.

[21] Rosin, P.L. and G.A.W. West (1990). Perceptual grouping of circular arcs under projection. *1st British Machine Vision Conf.*, Oxford, UK, 379–382.

[22] Safaee-Rad, R., I. Tchoukanov, B. Benhabib and K.C. Smith (1991). Accurate parameter estimation of quadratic curves from grey-level images. *Computer Vision, Graphics, and Image Processing: Image Understanding* 54, 259–274.

[23] Sampson, P. (1982). Fitting conic sections to 'very scattered' data: an iterative refinement of the Bookstein algorithm. *Computer Vision, Graphics, and Image Processing.*

[24] Shirai, Y. (1978). Recognition of real-world objects using edge cue. In: A.R. Hanson and E.M. Riseman, Eds., *Computer Vision Systems.* Academic Press, New York, 353–362.

[25] Tsuji, S. and F. Matsumoto (1978). Detection of ellipses by a modified Hough transform. *IEEE Trans. Comp.* 27, 777–781.

[26] Wang, R., A.R. Hanson and E.M. Riseman (1988). Fast extraction of ellipses. *Proc. 9th ICPR*, 508–510.

[27] West, G.A.W. and P.L. Rosin (1991). A multistage ellipse and line detection algorithm. *DICTA-91*, Melbourne, 66–73.

[28] Wu, J. and T. Caelli (1988). Model-based 3D object localization and recognition from a single intensity image. *Vision Interface Conf.*, Edmonton, Alberta, Canada, 21–67.