Superellipse Fitting to Partial Data

Xiaoming Zhang
IBM WebSphere MQ New Technologies Development
IBM UK Laboratories
Winchester, SO21 2JN
UK
Zhang@uk.ibm.com

Paul L. Rosin
Department of Computer Science
Cardiff University
UK
Paul.Rosin@cs.cf.ac.uk

Abstract

Superellipses can be used to represent in a compact form a large variety of shapes, and are useful for modelling in the fields of computer graphics and computer vision. However, fitting them to data is difficult and computationally expensive. Moreover, when only partial data is available the parameter estimates become unreliable. This paper attempts to improve the process of fitting to partial data by combining gradient and curvature information with the standard algebraic distance. Tests show that the addition of gradient information seems to enhance the robustness of fit and decrease the number of iterations needed. Curvature information appears to have only marginal effects.

Index Terms– superellipse, fitting, optimization, objective function, algebraic distance

1 Introduction

Image feature extraction is an important part of a computer vision system. The accuracy and reliability of the scheme used for representing the features can have a profound impact on the higher level processes built on top. Curves are a common feature type, and provide a convenient representation of edges consisting of individual pixels, allowing easy manipulation at later processing stages. An interesting abstraction of curves is the superellipse. Its equation (assuming the canonical position: centred on the origin with its axes aligned with the co-ordinate system) is

$$\left(\frac{x}{a}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon}} = 1 \tag{1}$$

and was discussed by Gabriel Lamé as long ago as 1818, but it was not until this century when it was popularised. This took the form of successive waves, starting with its introduction to design by the Danish artist/poet/scientist Piet Hein, and continuing onto recreational mathematics [1], computer graphics [2], and computer vision [3]. The advantage of the superellipse is that it provides a fairly compact parameterisation that still enables a wide variety of shapes to be represented. In fact, it has been claimed to "combine 'the best' of the circle, the square, the ellipse and the rectangle", all of which can be represented by (1). Some further examples of well known curves that can also be represented are the Astroid ($\epsilon = \frac{3}{2}$) and the Witch of Agnesi ($\epsilon = \frac{3}{2}$). Unfortunately the superellipse's disadvantage is the highly non-linear nature of ϵ.[1] In particular, fitting is complicated and problematic since iterative methods are required. But even just for the generation of curves difficulties arise, for instance in producing (approximate) equal-distance sampling [4, 5].

In this paper we are particularly interested in superellipse curve fitting, i.e., estimating its parameters from a given set of pixel data. Most commonly the least squares solution is found [6, 7, 8]. This is

---

[1]For rational exponents the curve is algebraic while irrational exponents produces transcendental curves.

achieved by finding the set of model parameters that minimise the sum of the squares of the *distances* between the model curve and given pixel data. The notion of distance can be interpreted in various ways and Rosin and West [9] investigate several examples. The Euclidean distance between a pixel and the point on the curve closest to it is probably the most suitable measure. Unfortunately finding this distance is generally very time consuming and therefore is rarely used. Even for ellipses it requires solving a quartic equation, while for superellipses we have no closed form solution. A simpler but still effective measure is the algebraic distance given by

$$Q_0(x, y) = \left(\frac{x}{a}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon}} - 1. \tag{2}$$

The best fit superellipse is determined by finding the parameters which minimise the objective function $Q_0^2(x, y)$. One of the main problems with the algebraic distance is that it results in different distance estimates for different parts of superellipse curves depending on the local curvature of the curve. Many distance measures have been suggested to compensate the inaccuracy but none of them are totally satisfactory [10]. The inaccuracies of the distance approximations are not such a serious problem when they are adopted for fitting superellipses with relatively complete pixel samples (i.e. large parts of the data are not occluded). This is because errors in parameter estimation caused by measuring pixels opposite to each other (with respect to the centre of the superellipse) tend to cancel each other.

Realistic applications frequently require fitting based on pixel data covering only parts of superellipses. In such cases, a conventional fitting approach using the conventional algebraic distance measure often leads to a gross misestimation. Figure 1 shows an example of such an error. It can be seen from the figure that, although the estimated superellipse has a centre far away from the true centre (it is off the page to the left), the estimated superellipse fits the visible pixels reasonably well. Unfortunately, such an estimation would be deemed unsatisfactory for many applications.

A reasonable mathematical explanation of this phenomenon is that, in these cases, the algebraic distance measure results in error functions with a very flat error surface (or even local minima) surrounding the global minimum. Thus it is hard for conventional hill-climbing estimation techniques to converge to the true parameters. A natural idea to attack this problem is to somehow recondition the error surfaces to make the global minimum easier to find. This can be achieved by incorporating extra "distance" information. In this paper, we propose a modification scheme based on introducing higher order derivative information and demonstrate experimentally that it does aid superellipse fitting to partial data. The proposed scheme still allows the distance measure to be calculated in closed form (although the overall superellipse fitting remains iterative) and is therefore computationally attractive.

2 Proposed Scheme

2.1 Rationale

A problem with fitting superellipses to partial data using the conventional algebraic distance is that this error landscape does not provide enough information to guide the optimisation process. The conventional algebraic distance measure treats pixels as individual data points and relations between pixels are not exploited. However, incorporating these relations is potentially very useful for curve fitting. In particular, the local gradient and curvature of a superellipse are functions of its parameters. Therefore, examining the gradient and curvature features of points on a curve should also reveal information about the true parameters. For example, a segment of curve containing a sharp corner (a few points with substantial curvature surrounded by many points with much lower curvature) indicates a very small ϵ.

In order to utilize such information, reasonable gradient and curvature "distance" measures need to be defined. The next section devises these distance measures, which take the same form as the algebraic distance measure. We demonstrate that these distance measures can be easily combined together and used in an iterative parameter estimation algorithm.

Needless to say, the generalised distance measure will result in a different error surface in the parameter space. The expectation is that by combining different sources of information the new error surface will be better adapted to fitting partial data than the error surface produced by the conventional distance measure.

2.2 Modified Objective Functions

For generality, we first adopt a more general form of the algebraic distance which includes superellipses not in the canonical position

$$Q_0(x, y) = \left[\frac{(x - x_c)\cos\theta - (y - y_c)\sin\theta}{a}\right]^{\frac{2}{\epsilon}} + \left[\frac{(y - y_c)\cos\theta + (x - x_c)\sin\theta}{b}\right]^{\frac{2}{\epsilon}} - 1 \qquad (3)$$

where (x_c, y_c) are the coordinates of centre of the superellipse and θ is the degree of rotation. To simplify the expression of the following derivation, we define the one-to-one transformation $T(x_c, y_c, a, b, \theta, \epsilon)$ from the original space to a new $\{X, Y\}$ space:

$$\begin{cases} X = \left[\frac{(x - x_c)\cos\theta - (y - y_c)\sin\theta}{a}\right]^{\frac{1}{\epsilon}} \\ Y = \left[\frac{(y - y_c)\cos\theta + (x - x_c)\sin\theta}{b}\right]^{\frac{1}{\epsilon}} . \end{cases} \qquad (4)$$

Notice that transformation T does not only shift, rotate, and rescale superellipses but also transforms them into circles (due to the effect of ϵ in the transformation) in the XY domain. As a result, we have the following relationship between an algebraic distance to a unit circle in the XY domain and the $Q_0(x, y)$ in the xy domain:

$$Q_0'(X, Y) = X^2 + Y^2 - 1 = Q_0(x, y) \qquad (5)$$

In other words, we have performed a normalisation to transform the problem to circle fitting in the XY domain. Since the inverse mapping $T^{-1}(x_c, y_c, a, b, \theta, \epsilon)$ exists and is one-to-one as well, the xy and XY notations are theoretically interchangable in mathematical expressions. In the rest of the paper, we will mainly base our discussion in the xy domain but use the XY notations in places for clarity.

The conventional algebraic distance is calculated by evaluating $Q_0(u, v)$, where (u, v) is the coordinates of the point in question. If point (u, v) is on the superellipse described by $Q_0(x, y) = 0$, then $Q_0(u, v) = 0$, in other words the algebraic distance from (u, v) to $Q_0(x, y) = 0$ is 0. Otherwise, $Q_0(u, v) \neq 0$. Hence, curve fitting can be achieved by minimising objective function $Q_0^2(x, y)$.

The principle contribution of this paper is to extend the concept of algebraic distance to incorporate a measure of the local shape of the curve in terms of gradient, curvature, etc. In order to achieve this, we first look at the unit circle in XY space. For any point on the unit circle, (U, V), we have

$$U^2 + V^2 = 1, \qquad (6)$$

and (U, V) can be expressed in parametric form as

$$\begin{cases} U = \cos t \\ V = \sin t. \end{cases} \qquad (7)$$

By differentiating both sides of equation (6) with respect to t we have

$$\frac{d(U^2 + V^2)}{dt} = 2UU_t' + 2VV_t' = 0. \qquad (8)$$

Similarly,

$$\frac{d(UU_t' + VV_t')}{dt} = [UU_t'' + (U_t')^2 + VV_t'' + (V_t')^2] = 0. \qquad (9)$$

Notice that from Equation (7) we get $(U_t')^2 + (V_t')^2 = 1$, giving

$$(UU_t'' + VV_t'' + 1) = 0. \qquad (10)$$

Let $Q_1(x, y) = XX_t' + 2YY_t'$ and $Q_2(x, y) = XX_t'' + YY_t'' + 1$. Thus, in a similar manner to the algebraic distance, at any point, (u, v), on the superellipse $Q_0(u, v) = Q_1(u, v) = Q_2(u, v) = 0$. In general, for any point off the superellipse $Q_1(x, y) \neq 0$ and $Q_2(x, y) \neq 0$, although depending on the values of X_t', Y_t', X_t'' and Y_t'' this may not hold. In contrast, Q_0 is guaranteed to be non-zero for all points off the superellipse.

The above suggests that we can introduce Q_1 and Q_2 into our objective function for minimisation. One way to achieve this is to use a linear combination. For example, we can define a new objective function in the XY domain (although written as $F(x, y)$)

$$F(x, y) = (1 - w_1 - w_2)Q_0^2(x, y) + w_1 Q_1^2(x, y) + w_2 Q_2^2(x, y) \quad (w_1, w_2 \geq 0 \text{ and } w_1 + w_2 \leq 1) \quad (11)$$

3

so that the curve fitting problem is now to minimise $F(x, y)$, which involves simultaneously minimising $Q_0^2(x, y)$, $Q_1^2(x, y)$, and $Q_2^2(x, y)$.

The purpose of the objective function is to measure how well the gradient and curvature of a given data set fits a theoretical superellipse model. However there is no need for us to explicitly calculate the theoretical values of X_t', Y_t', X_t'', and Y_t''. All that is needed is to estimate them from the data set and evaluate Q_1 and Q_2 accordingly.

It should be pointed out that the reduction of $F(x, y)$ does not necessary lead to the reduction of a given distance measure (say, the Euclidean distance) in the xy domain. However, an exact fit in the XY domain ($F(x, y) = 0$) does guarantee an exact superellipse fit in the xy domain.

2.3   Application of the Levenberg-Marquardt Algorithm

The problem of curve fitting is achieved by finding a set of parameters which minimises an objective function. As pixel data are given, x and y are no longer treated as variables during parameter estimation. Instead, F should be expressed as functions of the unknown parameters. For the superellipse model given earlier, there are six parameters, namely, x_c, y_c, a, b, θ, and ϵ. Our objective function, therefore, should be expressed as $F(x_c, y_c, a, b, \theta, \epsilon)$ and it is non-linear.

To minimise the non-linear objective function, we have applied the well-known Levenberg-Marquardt iterative algorithm [11]. This has become a standard algorithm for non-linear optimisation due to its robustness and efficiency obtained by a clever combination of the steepest descent method and Newton's algorithm. The former have rapid convergence but tend to overshoot unless provided with a good initialisation. On the other hand, Newton's Hessian based method initially has slow convergence, but speeds up closer to the minimum. In the Levenberg-Marquardt algorithm, when the results are far from the minimum the method of steepest descent essentially determines the step size. As the solution approaches the minimum the Hessian matrix has more effect on determining the step size.

The Levenberg-Marquardt algorithm requires the analytic expressions of $\frac{\partial F}{\partial p_i}$ and $\frac{\partial^2 F}{\partial p_i \partial p_j}$, p_i being the six parameters. As $\frac{\partial X_t'}{\partial p_i} = \frac{\partial Y}{\partial p_i}$, $\frac{\partial Y_t'}{\partial p_i} = -\frac{\partial X}{\partial p_i}$, $\frac{\partial X_t''}{\partial p_i} = -\frac{\partial X}{\partial p_i}$, and $\frac{\partial Y_t''}{\partial p_i} = -\frac{\partial Y}{\partial p_i}$,

$$
\begin{aligned}
\frac{1}{2}\frac{\partial F}{\partial p_i} = {} & (1 - w_1 - w_2)Q_0 \left(X\frac{\partial X}{\partial p_i} + Y\frac{\partial Y}{\partial p_i} \right) + \\
& w_1 Q_1 \left[(X_t' - Y)\frac{\partial X}{\partial p_i} + (Y_t' + X)\frac{\partial Y}{\partial p_i} \right] + \\
& w_2 Q_2 \left[(X_t'' - X)\frac{\partial X}{\partial p_i} + (Y_t'' - Y)\frac{\partial Y}{\partial p_i} \right].
\end{aligned}
\tag{12}
$$

One might think that the terms $X_t' - Y$ and $Y_t' + X$ on the right hand side of the equation are always zero (as $X_t' = Y$ and $Y_t' = -X$) and therefore part of the right hand side is redundant. However, when evaluating the right hand side (X_t', Y_t') is derived from (X, Y) which in turn is worked out by using the current estimate of the parameter set $(x_c, y_c, a, b, \theta, \epsilon)$. Equations $X_t' - Y$ and $Y_t' + X$ will be zero only when the true $(x_c, y_c, a, b, \theta, \epsilon)$ is found.

It is a convention that second derivatives can be omitted in the Levenberg-Marquardt iteration [11]. Therefore,

$$
\begin{aligned}
\frac{1}{2}\frac{\partial^2 F}{\partial p_i \partial p_j} \approx {} & (1 - w_1 - w_2) \left(X\frac{\partial X}{\partial p_i} + Y\frac{\partial Y}{\partial p_i} \right)\left(X\frac{\partial X}{\partial p_j} + Y\frac{\partial Y}{\partial p_j} \right) + \\
& w_1 \left[(X_t' - Y)\frac{\partial X}{\partial p_i} + (Y_t' + X)\frac{\partial Y}{\partial p_i} \right]\left[(X_t' - Y)\frac{\partial X}{\partial p_j} + (Y_t' + X)\frac{\partial Y}{\partial p_j} \right] + \\
& w_2 \left[(X_t'' - X)\frac{\partial X}{\partial p_i} + (Y_t'' - Y)\frac{\partial Y}{\partial p_i} \right]\left[(X_t'' - X)\frac{\partial X}{\partial p_j} + (Y_t'' - Y)\frac{\partial Y}{\partial p_j} \right].
\end{aligned}
\tag{13}
$$

Derivation of $\frac{\partial X}{\partial p_i}$ and $\frac{\partial Y}{\partial p_i}$ is straightforward, and is therefore omitted.

In each Levenberg-Marquardt iteration, X_t', Y_t', X_t'', and Y_t'' are re-estimated from the transformed data (i.e. (X, Y)) in order to evaluate $\frac{\partial F}{\partial p_i}$. Currently, (X_t', Y_t') is estimated by fitting a line to (X, Y) using linear regression [12]. This introduces another parameter, namely, the width of the observation window. The optimal window size is expected to depend on the noise level of the input data. It is possible to implement the estimation in an $O(N + W)$ fashion, N being the number of pixels and W the

width of window. In the same way (X_t'', Y_t'') is estimated by fitting a line to (X_t', Y_t').[2]

Compared to fitting just the standard algebraic distance Q_0 the total time complexity for each Levenberg-Marquardt iteration only increases by a small constant. Of more importance, preliminary tests on partial data show that introducing the higher order features (particularly the gradient information) dramatically reduces the number of iterations required for the Levenberg-Marquardt algorithm to converge, providing an overall speedup.

Providing good initial guesses of the model parameters for the Levenberg-Marquardt algorithm should reduce the amount of Levenberg-Marquardt iterations required. We use the least-squares ellipse fit to the data [13] to provide estimates for x_c, y_c, a, b, and θ. The initial guess of ϵ is always set to 0.5, the centre point of the range of ϵ we consider. This procedure is efficient (its time complexity is linear in the number of points) and provides reasonable estimates for most of the parameters.

3 Experimental Tests

3.1 Test data generation

In order to carry out extensive tests, 4000 synthetic data sets were used, each containing around 1000 points. Some examples are shown in figure 2. As an initial attempt, it is sensible to focus our attention only to some of the possible tests. The following specify the extent of this investigation.

Added noise − In practice, errors in pixel data are inevitable due to sampling error, noise existing in original image, etc. To simulate this, the procedure for generating data is as follows:

1. Perfect superellipses are synthesised for different instances of the parameter values. The centre is fixed at $(x_c, y_c) = (400, 400)$, and $b = 100$. The remaining parameters are evenly sampled:
    - a values in range: $[150, 260]$ pixels, increment: 10
    - ϵ values in range: $[0.1, 1.0]$, increment: 0.1
    - θ values in range: $[0, 3]$ radians, increment: 0.3

    The co-ordinate data (in the range $[0, 800]$) is quantised to integer form and plotted in an image.

2. The interior of the superellipses is flood-filled and the resulting binary image is blurred slightly by Gaussian smoothing ($\sigma = 2$).

3. Gaussian noise ($\sigma = 20$) is added to the image pixel values.

4. The image is thresholded, and the noisy superellipse boundary extracted.

Magnitude of ϵ − It is not difficult to appreciate that a particular curve segment can frequently be approximated by two superellipses with two different values of ϵ, one larger than 1 and the other less than 1. To simplify and speed up testing we only generate $\epsilon < 1$.

Portion of superellipse used and its location − Tests have shown that both the standard and new least squares schemes do not perform very well when, for $\epsilon < 1$, less than two (rounded) corners are present in the data. We believe that this is because the data surrounding the corners provides the majority of the information concerning the shape of the superellipse. We set the amount of data present to 60% (i.e. a subtended angle of 216°) to ensure that a reasonable amount of corner information is available.

Variation of parameter a and b − It is our believe that the performance of the scheme will remain roughly the same as long as a/b remains constant. We therefore fix the value for b to 100 and only vary a when generating superellipses.

3.2 Test setup

The performance of this proposed scheme can be affected by many algorithm parameters such as the values of w_1 and w_2. Four aspects are outlined below.

---

[2]An alternative would be to estimate the curvature by fitting a circle directly to the curve segment.

Weights w_1 and w_2 – Weights w_1 and w_2 can be set to various possible values. As gradient and curvature features are estimated values which contain error, large w_1 and w_2 values are unlikely to be appropriate. Here, we only investigate w_1 and w_2 within the range of $0 \leq w_1, w_2 \leq 0.3$.

Selection of window sizes for gradient and curvature estimation – To estimate local gradient and curvature features at each point, it is necessary use additional neighbouring points in addition to the central point. Setting the optimal window size is problematic – the presence of noise makes small windows unreliable while large windows distort the local line fitting. After some investigation, we have set the window size for gradient and curvature to 101 which corresponds to $5 - 10\%$ of the curve.

Dealing with unevenness of pixel distribution – Definition (4) can be viewed as a transformation from the input xy domain to the XY domain. Assuming the original pixels in xy are evenly distributed along superellipse curves, the transformed pixels in XY are unlikely to be distributed evenly [4, 5]. This means that mid-point pixels may not actually be at the centre of gradient/curvature estimation windows, which may result in estimation errors. Currently, no effort has been made to remedy the situation, assuming the imbalance is insignificant.

Estimation of gradient and curvature at segment ends – Estimation of gradient and curvature requires multiple neighbouring pixels. At both two ends of data segments, some neighbouring pixels are not available. In general, there are two ways to solve this problem during curve fitting. The first one is to ignore all pixels which have insufficient neighbouring pixels. This may result in the loss of important information. The other one is to estimate using information available. We have chosen to duplicate the nearest end pixel to replace missing pixel neighbours.

3.3 Test results

Examples of fitting to the data are shown in figure 2. The extreme errors that often arise are evident. A more quantitative assessment is provided by the following graphs. They show the quality of the various parameter estimates as the parameters of the data are varied as well as the parameters of the algorithm. The centre errors are calculated as the Euclidean distances between the true and the estimated centre co-ordinates, and the orientation error is measured in radians. The general performance of the algorithm is tested on the full data set of 4000. Further testing of specific variations in the algorithm or data was carried out on subsets of between 750 and 1500 data sets.

General performance

The first set of tests was done using a window size of 101 for gradient and curvature estimation. Figure 3 shows the test results for four different w_1/w_2 combinations. They represent tests on all the test data generated using the various model parameters shown on the abscissa. The two rows display the estimation errors of the parameters as functions of a and ϵ. Assuming b fixed, these are the only shape parameters as opposed to pose parameters. Several observations can be made:

- Overall, it is evident that the introduction of the gradient/curvature features has significantly reduced the estimation error of all the individual parameters.

- Estimation error of a generally increases as a (or ratio a/b) increases. This means that superellipses with a large aspect ratio are more difficult to fit.

- The estimation error of θ generally decreases as a increases. This is easy to understand as more elongated superellipses have a better defined orientation – the orientation of circles can not be estimated.

- For low squareness values ($\epsilon \in [0, 0.5]$) – i.e. more rectangular shapes – the inclusion of gradient information/curvature improves the estimation accuracy, while for larger squareness values ($\epsilon \in [0.6, 1]$) – i.e. more elliptical shapes – estimation accuracy is reduced.

- There is an inherent dependency between the estimated major axis a and the center, which is evident in these figures.

Impact of w_1 and w_2 variation

A second set of tests have been carried out to reveal the effect of varying w_1 and w_2. Figures 4 and 5 show the performance as a function of one of the weights while the other is kept fixed. It appears that the introduction of a small amount of gradient information (w_1) improves performance – errors are reduced by a factor of two or three. Increasing w_1 beyond ≈ 0.1 produces no further benefit. Including curvature (w_2) in addition to gradient information makes little difference, and no across the board improvement for all the parameter estimates was found.

Impact of window size variation

Figure 6 shows the effect of varying the size of the window within which the line fitting is performed to estimate the gradient and curvature information. It is seen for this data set that the best performance is achieved around 100 (which corresponds to $5-10\%$ of the curve) although we would expect this to alter for different data sets. It can be seen that greater errors are incurred by too large a window (resulting in blurring of the corners) rather than too small a window (insufficient noise suppression).

Impact of noise level variation

Since the proposed method requires the estimation of gradient and curvature information there is the concern that it will be overly sensitive to noise. Figure 7 shows the effect of modifying the noise level in the data generation procedure. As expected, adding large amounts of noise does increase the error rate. However, the algebraic distance augmented by the gradient and curvature information retains higher accuracy of the parameter estimates.

Impact of variation in completeness of data

We previously showed that fitting superellipses using the algebraic distance appears to breakdown when less than 60–70% of the data is present [10]. Figure 8 shows the results of testing the standard and augmented algebraic distances. It can be seen that the standard algebraic distance performs better as long as 70% or more of the data is present. When less than 70% is available all fits noticeably degrade, but those using the gradient information do not exhibit such a sharp breakdown, and consequently provide consistently better estimates.

Impact on efficiency

Earlier we hypothesised that augmenting the algebraic distance could improve the shape of the error surface, and thereby aid convergence. This is verified by figure 9 which shows that the number of iterations is considerably reduced. Only a small amount of gradient information ($w_1 = 0.05$) is necessary to halve the number of iterations. Despite the overhead of the additional gradient calculations the overall computation time is still reduced.

3.4   Fitting to Real Data

We finish with an example of applying the fitting techniques to real data. A set of edges are extracted from the image in figure 10. The results of the fits are overlaid on the data, and plotted both when the standard algebraic distance is used (figure 11) and when gradient information is incorporated (figure 12). It can be seen that about half of the estimated superellipses are similar for both methods, while most of the remaining fits are better when gradient information is used. The majority of these correspond to data which are short or have low curvature, and the algebraic distance fits tend to overestimate the value of the major axis (compare the high curvature bias problem that arises with ellipse fitting [14]).

3.5   Application to Classification

A final test of the fitting technique is given in a classification example. Figure 13 shows a variety of seeds and beans to be distinguished; the complete data set contained 260 samples with between 25 and 30 samples for each of the nine classes. The resolution is relatively low: each sample is between 20 and 50 pixels in length. To make the process scale invariant only intrinsic shape parameters are used. Since the seeds and beans have similar shapes and moreover display considerable intra-class variability this is a difficult classification task. Previously, we performed classification on a similar data set using Murthy *et*

al.'s oblique decision trees [15] applied to various shape measures [16]. Rerunning the experiment using the best performing two shape measures (ellipticity and rectangularity as defined in [16]) the classifier gave 53.85% accuracy using 10 fold cross-validation. Including squareness as determined from the superellipse fits ($w_1 = 0.1$ and $w_2 = 0$) boosted performance to achieve 57.31%.

4 Conclusions

Reliably fitting superellipses to partial data is both difficult and computationally expensive, requiring iterative techniques to minimise some objective function. We have described an approach that augments the standard algebraic distance commonly used as the objective function to incorporate gradient and curvature information. It was found that by adding just a small amount of gradient information to the algebraic distance both the quality of the fit was improved and the computation time was decreased. Tests showed this to hold i) over a range of noise levels, and ii) under conditions where less than 70% (i.e. a subtended angle of 252°) of the data is present.

References

[1] M. Gardiner, The superellipse: a curve that lies between the ellipse and the rec tangle, Scientific American 21 (1965) 222–234.

[2] A. Barr, Superquadrics and angle-preserving transformations, IEEE Computer Graphics Applications 1 (1981) 11–23.

[3] A. Pentland, Automatic extraction of deformable part models, Int. J. Computer Vision 4 (2) (1990) 107–126.

[4] W. Franklin, A. Barr, Faster calculation of superquadric shapes, IEEE Computer Graphics Applications 1 (1981) 41–47.

[5] M. Pilu, R. Fisher, Equal-distance sampling of superellipse models, in: British Machine Vision Conf., 1995, pp. 257–266.

[6] A. Gross, T. Boult, Error of fit for recovering parametric solids, in: Proc. Int. Conf. Computer Vision, 1988, pp. 690–694.

[7] F. Solina, R. Bajcsy, Recovery of parametric models from range images: the case for superquadrics with global deformations, IEEE Trans. on Pattern Analysis and Machine Intelligence 12 (1990) 131–147.

[8] N. Yokoya, M. Kaneta, K. Yamamoto, Recovery of superquadric primitives from a range image using simulated annealing, in: Int. Conf. Pattern Recognition, 1992, pp. 168–172.

[9] P. Rosin, G. West, Curve segmentation and representation by superellipses, Proc. IEE: Vision, Image, and Signal Processing 142 (1995) 280–288.

[10] P. Rosin, Fitting superellipses, IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (7) (2000) 726–732.

[11] W. Press, B. Flannery, S. Teukolsky, W. Vettering, Numerical Recipes in C, Cambridge University Press, 1990.

[12] R. Haralick, L. Shapiro, Computer and Robot Vision, Addison Wesley, 1992.

[13] A. Fitzgibbon, M. Pilu, R. Fisher, Direct least squares fitting of ellipses, in: Proc. Int. Conf. Pat. Recog, 1996, pp. 253–257.

[14] P. Rosin, Assessing error of fit functions for ellipses, Graphical Models and Image Processing 58 (1996) 494–502.

[15] S. Murthy, S. Kasif, S. Salzberg, System for induction of oblique decision trees, Journal of Artificial Intelligence Research 2 (1994) 1–33.

[16] P. Rosin, Measuring shape: Ellipticity, rectangularity, and triangularity, in: Int. Conf. Pattern Recognition, 2000, pp. Vol I: 952–955.
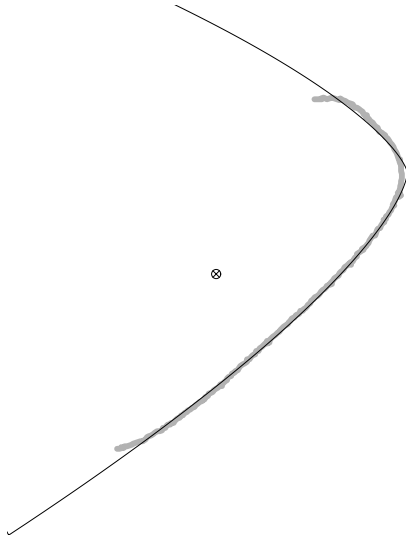
List of Figures

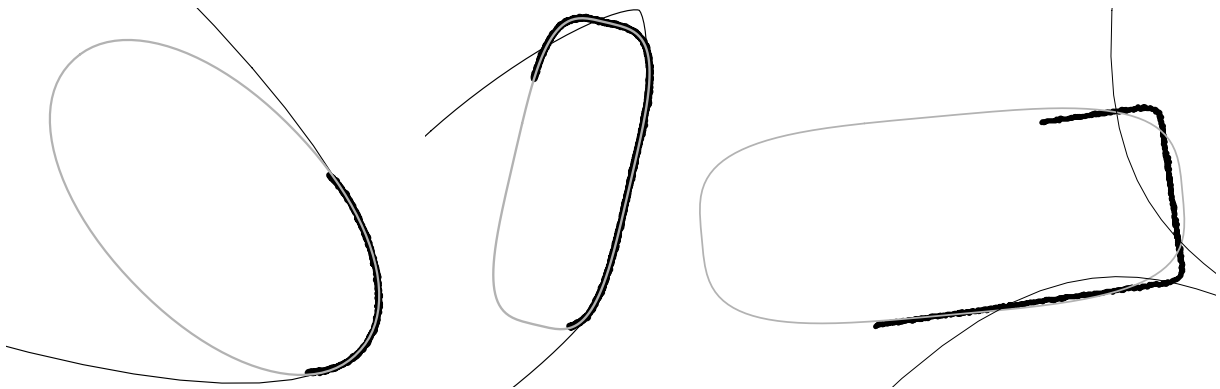Figure 1: Gross misestimation of centre

Figure 2: Examples of superellipse fits to partial data; the fit using the standard algebraic distance ($w_1 = 0$ and $w_2 = 0$) is drawn in black, the fit incorporating gradient information is drawn in gray ($w_1 = 0.1$ and $w_2 = 0$)
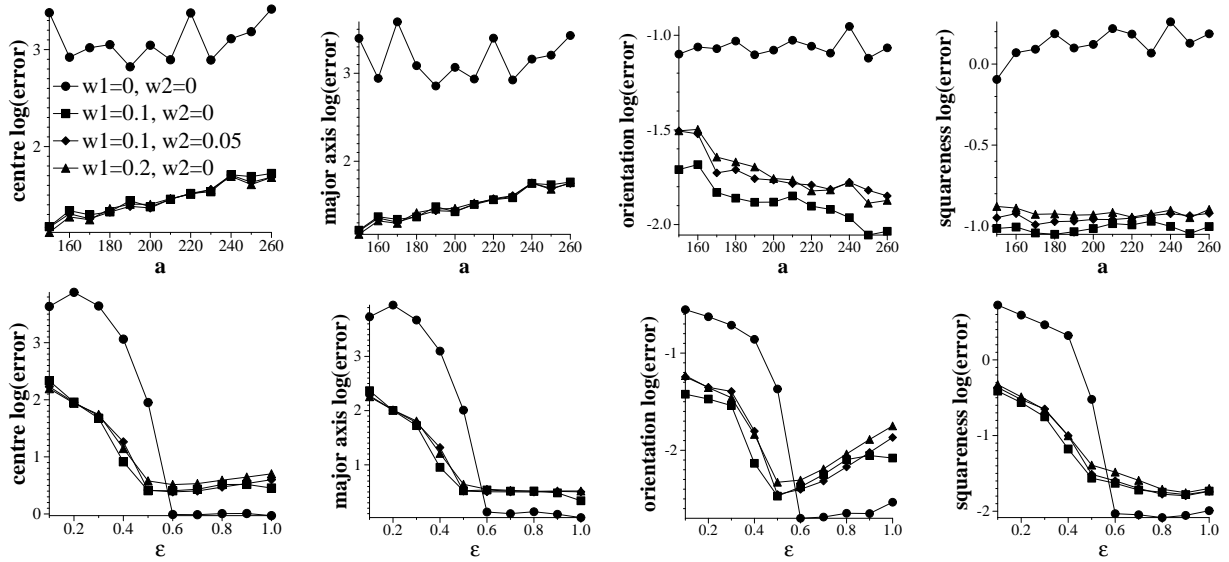
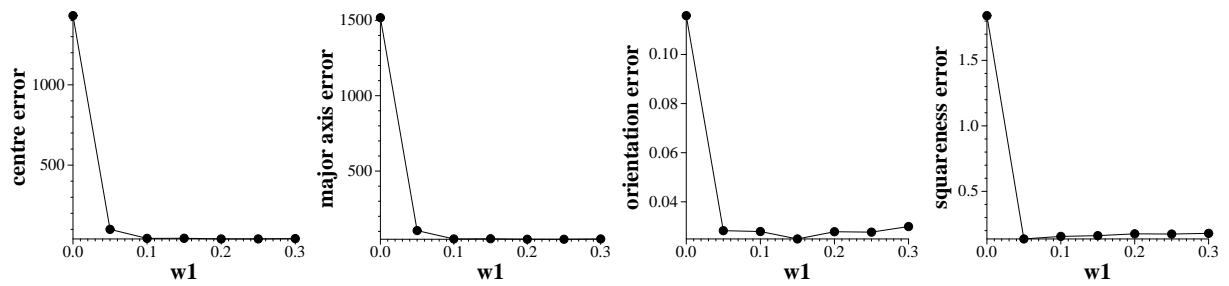Figure 3: Mean errors in parameter estimates

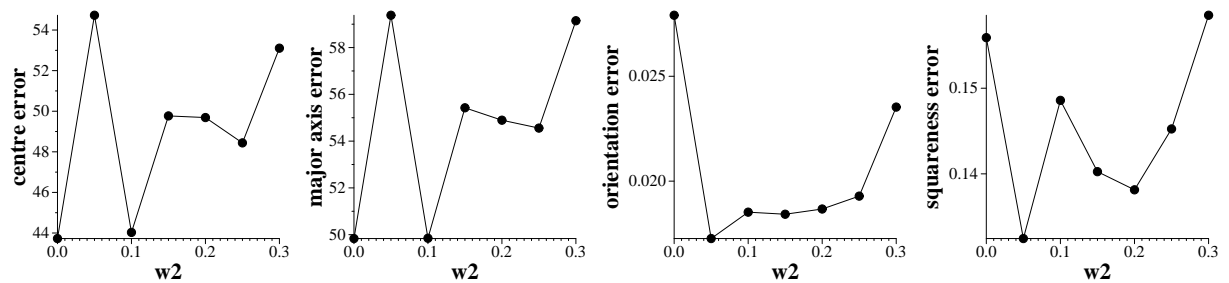Figure 4: Effect of varying weight w_1 with $w_2 = 0$
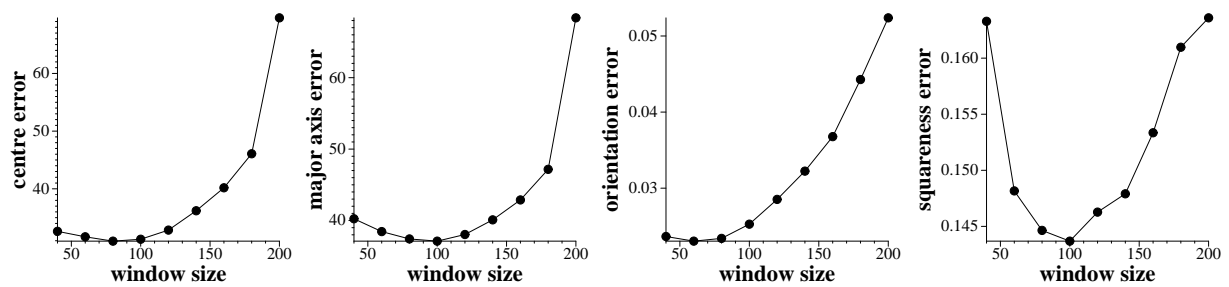

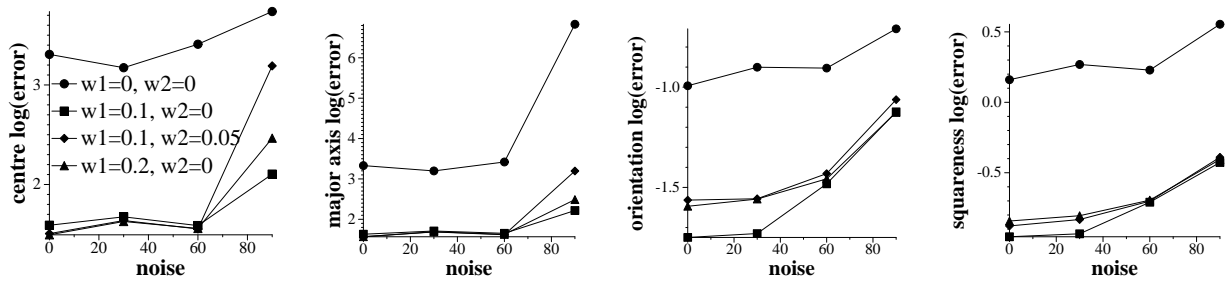Figure 5: Effect of varying weight w_2 with $w_1 = 0.1$

Figure 6: Effect of varying window size

Figure 7: Effect of varying noise levels
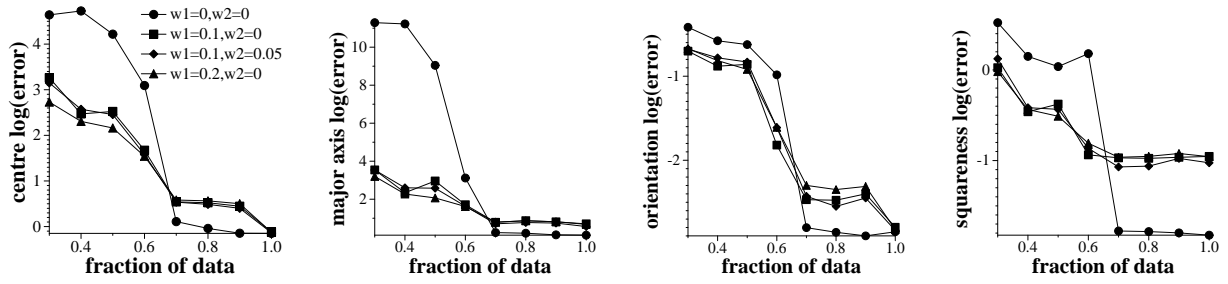
Figure 8: Effect of varying amounts of missing data
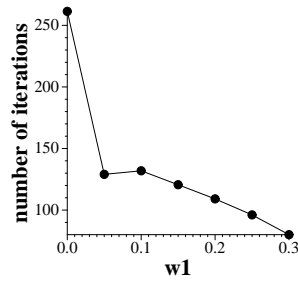
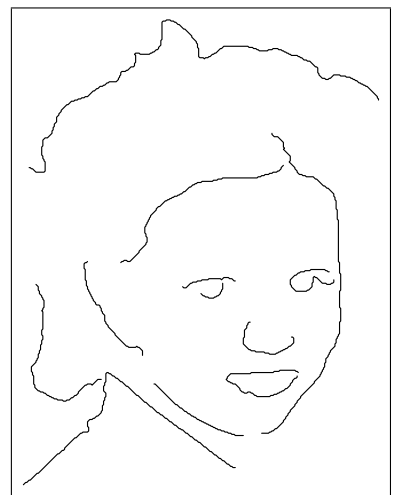Figure 9: Reduction in mean number of iterations when gradient information is included

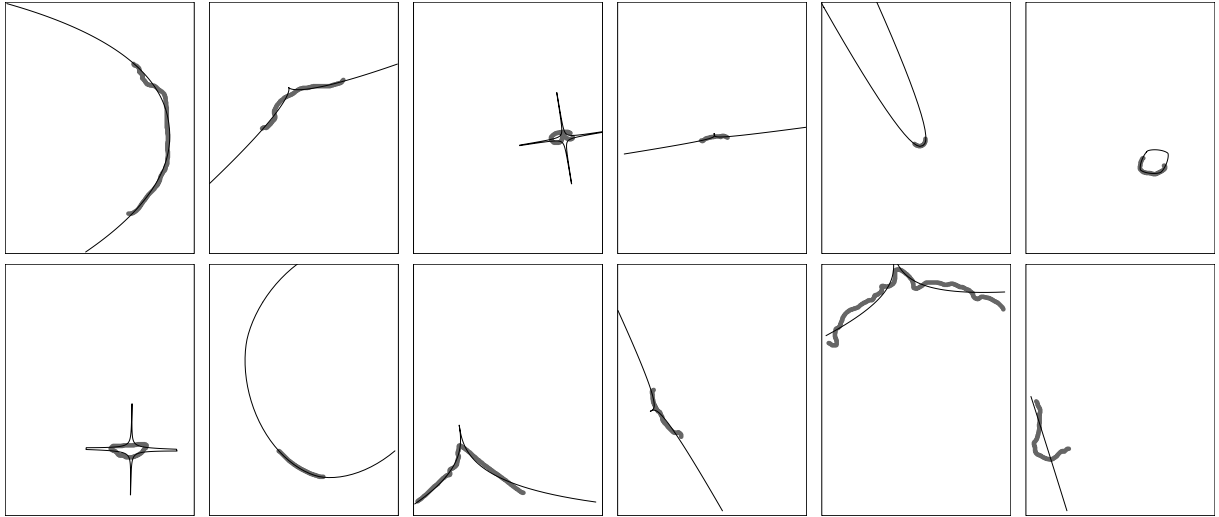Figure 10: Test image with extracted contours

Figure 11: Fitted superellipses without using gradient information: $w_1 = 0$ and $w_2 = 0$
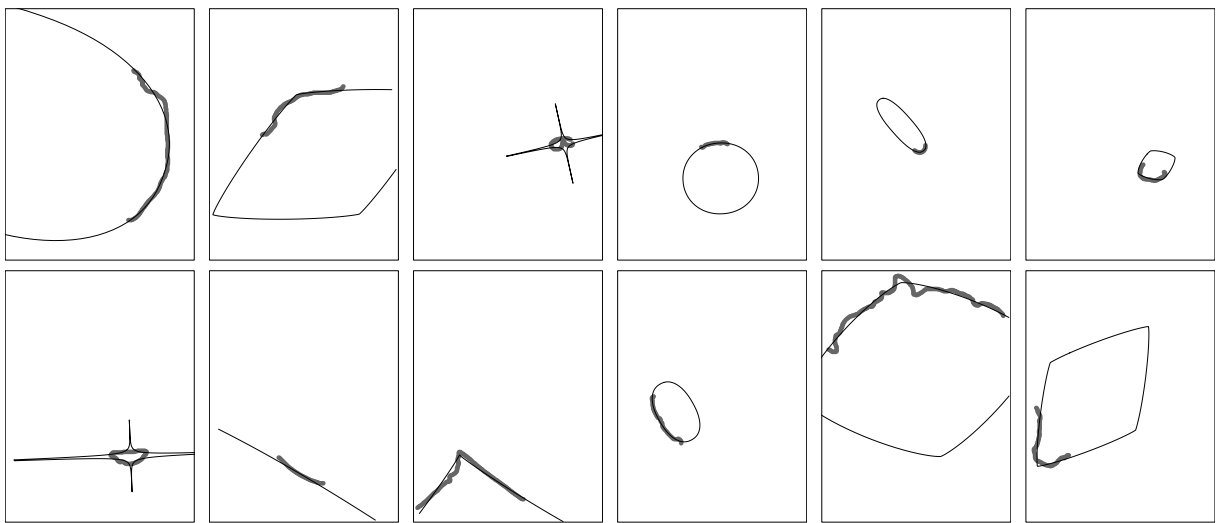

Figure 12: Fitted superellipses using gradient information: $w_1 = 0.05$ and $w_2 = 0$
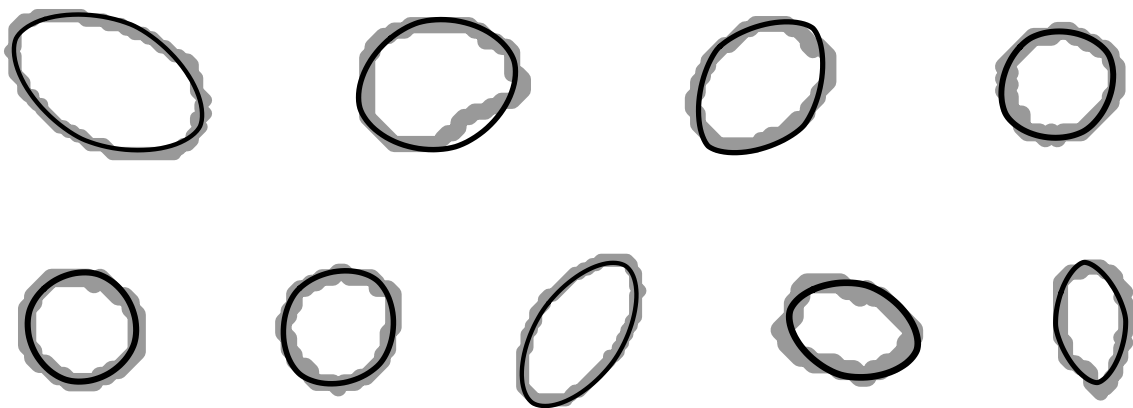

Figure 13: Examples of seed and bean data with fitted superellipses overlaid; (1) almond, (2) chickpea, (3) coffee bean, (4) lentil, (5) peanut, (6) corn kernel, (7) pumpkin seed, (8) raisin, (9) sunflower seed.

13