Curve Segmentation and Representation by Superellipses

Paul L. Rosin[1] and Geoff A .W. West[2]

[1]Environmental, Mapping and Modelling Unit,
Joint Research Centre,
Institute for Remote Sensing Applications,
I-21020 Ispra (Va) Italy.
email: paul.rosin@jrc.it

[2]Department of Computer Science,
Curtin University of Technology,
Bentley, Perth, Western Australia.
email: geoff@cs.curtin.edu.au

Abstract

A method of segmenting curves into series of superelliptical arcs is presented. A superellipse is the 2-dimensional form of the superquadric and can describe circles, ellipses, crosses, parallelograms and rounded rectangles with the same number of parameters. The superellipses are fitted using Powell's technique to minimise an appropriate error metric. A tree is used to represent a number of interpretations and the concept of significance used to choose the most perceptually correct description. Results show that perceptually good features are chosen to represent the various shapes that occur in images.

1 Introduction

A key problem in computer vision is the extraction of meaningful features from edge data. Although edge pixel data is useful in itself, it is better to represent edges in a more manageable form. The type of description required can be very application dependent but is usually based on a combination of straight line approximations and higher order curves. Numerous techniques have been proposed for generating straight line approximations of curve data. However, less time has been devoted to extracting higher order representations because of the increased number of parameters or degrees of freedom and the ill-conditioned

nature of the problem. In addition, most of the polygonal approximation techniques are specific in nature, and cannot be easily generalised to other features or data types. Lowe proposed an algorithm for segmenting curves into straight lines [13] which we have shown is easily generalised to fit a variety of representations such as circular arcs [25], elliptical arcs [19], and second-order polynomials [18]. Here we extend the algorithm to segment curves into superellipses.

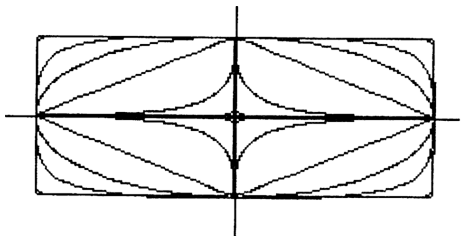


Figure 1: Various superellipses generated for constant values of a and b and different values of ε. Large values ε produce a cross, small values produce a rectangle.

Superellipses were first formulated by Peit Hein [8], and they and their three dimensional equivalent — superquadrics — were popularised in computer graphics by Barr [4] and computer vision by Pentland [14]. They are a flexible representation, and with a relatively small number of parameters can represent a wide variety of shapes including ellipses, rectangles, parallelograms, and pinched diamonds. Superellipses can be described in the parametric form by

$$\begin{aligned} x(t) &= a\cos^{\varepsilon} t \\ y(t) &= b\sin^{\varepsilon} t \end{aligned} \tag{1}$$

where the lengths of the axes are given by a and b and the *squareness* is determined by ε. To avoid evaluating complex roots equation 1 can be modified to

$$\begin{aligned} x(t) &= a\,\mathrm{sgn}\cos^{\varepsilon} t|\cos^{\varepsilon} t| \\ y(t) &= a\,\mathrm{sgn}\sin^{\varepsilon} t|\sin^{\varepsilon} t| \end{aligned} \tag{2}$$

Figure 1 shows superellipses with $a = 2b$ and ε set to 0.1, 0.5, 1.0, 2.0, 5.0 and 10.0. A value of $\varepsilon = 1$ produces an ellipse, $\varepsilon = 2$ produces a diamond, $\varepsilon > 2.0$ produces pinched diamonds (very large values result in a cross) and $\varepsilon < 1.0$ produces rectangles with rounded corners (very low values result in perfect rectangles).

2 The Segmentation Algorithm

Lowe [13] proposed a technique for segmenting curves into straight lines. We have generalised and extended this method to apply to other features and to combinations of different features. The method is based on recursive subdivision

of the curve to form a tree data structure which is traversed to select the best representation of the curve.

To segment a curve consisting of a list of pixels into features of a single type the complete pixel list is approximated by the required feature. The list is then split into two at the point of maximum deviation between the approximation and the underlying pixel data. This process of approximation and splitting is repeated recursively on each of the two lists and halts when the pixel lists are too small to properly fit the representation. Each approximation is stored in a data structure with links to its two split approximations. The result of the recursive process is a *segmentation tree* in which each level describes the complete curve by increasingly fine segmentations and approximations.

Each feature primitive in the segmentation tree is assigned a significance value. This was defined by Lowe as the ratio of the length of the feature divided by the maximum Euclidean deviation of the feature from the pixels. That is, for a feature primitive f_t (parameterised by arclength t) and a set of pixels p_i the deviation is

$$D(f_t, p_i) = \max_t \min_i \parallel f_t - p_i \parallel .$$

The length of the feature $L(f_t)$ is taken as the section of the feature which covers the pixel data. If the fitting process constrains the feature to pass through the pixel list endpoints then these delimit the feature. Otherwise the points on the feature which minimise the minimum distance to the pixel list endpoints are taken as the feature endpoints. To avoid divide by zero errors we have inverted Lowe's significance ratio giving

$$S(f_t, p_i) = \frac{D(f_t, p_i)}{L(f_t)}$$

which means that the feature selection process should now minimise the significance values.

The final segmentation is then found by searching the segmentation tree for the best (most significant) set of feature primitives (by choosing those with the lowest significance). Traversing up the tree from the leaves, a feature primitive is retained if it is more significant than all its children, otherwise it is replaced by its children. After traversing the tree the remaining feature primitives constitute the best segmentation of the curve.

When forming the segmentation tree the stopping criterion is that it must be possible to adequately fit a feature primitive to the subset of pixels. The fit should not be underdetermined (i.e. at least seven points are required for a superellipse). Also, it should not be possible to form a perfect fit between the feature and pixel data since this would result in a perfect significance value preventing small sections in the segmentation tree being replaced by larger and highly significant superellipses. Thus, for a superellipse, the minimum number of points that can be fitted to is eight, ensuring that the fitting process is overdetermined and produces superellipse fits with a finite error producing meaningful significances.

It is important that our stopping criterion is not confused with the error tolerance threshold used by standard recursive curve splitting algorithms which do not use backtracking (e.g. [7]). Whereas our criterion is fixed for each feature type being fitted, the threshold for the standard algorithms is data dependent, and must be carefully selected.

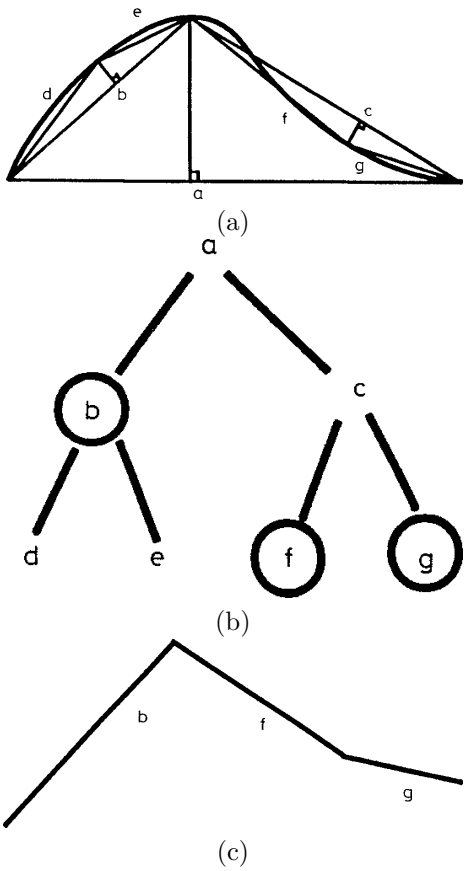


Figure 2: (a) A curve and the approximating straight lines and required distances, (b) segmentation tree — chosen representations circled, and (c) final choice of representations from the tree.

In figure 2 the process is illustrated with an example of segmenting a curve into straight lines. Figure 2a shows a curve and the complete set of approximating lines with their perpendiculars showing the points of maximum deviation between the curve and line. The generated segmentation tree is shown in figure 2b. The circled nodes indicate those selected as the most significant in the tree. That is, b is more significant than both d and e; either f or g is more significant than c; either b, f or g is more significant than a. The final curve segmentation is shown in figure 2c.

This segmentation technique has several desirable properties. First, no parameters are required throughout the process. In contrast, most other algorithms require thresholds for determining which points should be considered as potential breakpoints, and how small the error between the approximation and data must be before the segmentation process terminates. This applied not only to heuristic algorithms but also to statistical approaches Second, the segmentation process is combined with the fitting process so that the fitted representation is used to determine the breakpoints. This means that the approach can be readily extended to any type of representation that can be fitted. In contrast, many methods determine breakpoints by techniques specific to a single representation. For instance, a common procedure for segmenting curves into straight lines is to search for curvature extrema. However, more complex representations such as superellipses have complex curvature signatures, making them difficult to analyse. Third, no noise models are required, such as are common in statistical and minimum description length approaches. A weakness of these approaches is that the simple noise model (usually Gaussian) does not always apply, causing the techniques to give poor results in the presence of outliers.

3 Fitting Superellipses

Superellipses are fitted by finding the set of parameters that minimise some error measure between the curve and data, and the arc sections that cover the data are retained. The fits are not constrained to pass through the endpoints of the pixel lists. Therefore the endpoints of a superellipse arc are determined as the points on the arc which minimise the minimum distance to the pixel list endpoints. The mean Euclidean distance is the optimum metric which can be determined from the parametric form shown in equation 1. The x and y components of the normal to the superellipse in parametric form at angle t are:

$$\begin{aligned} x'(t) &= b\varepsilon \cos t \sin^{\varepsilon-1} t \\ y'(t) &= a\varepsilon \sin t \cos^{\varepsilon-1} t \end{aligned} \tag{3}$$

The gradient is then:

$$m = \frac{y'(t)}{x'(t)} = \frac{a \sin t \cos^{\varepsilon-1} t}{b \cos t \sin^{\varepsilon-1} t} \tag{4}$$

The value of t for the nearest point on the superellipse (x_e, y_e) can be determined using equation 1 and

$$m = \frac{y_p - y_e}{x_p - x_e} \tag{5}$$

to give:

$$\frac{a \sin t \cos^{\varepsilon-1} t}{b \cos t \sin^{\varepsilon-1} t} = \frac{y_p - a \cos^{\varepsilon-1} t}{x_p - b \sin^{\varepsilon-1} t} \tag{6}$$

which enables the distance of the point to the ellipse d to be determined from:

$$d = \sqrt{(y_p - y_e)^2 + (x_p - x_e)^2} \tag{7}$$

Unfortunately it is difficult to solve equation 6 for t analytically although the value of t that minimises d can be determined numerically by walking around the superellipse. This would be computationally expensive. Instead, alternative simpler measures, similar to those used for ellipse fitting [17] and polynomial fitting [22] were investigated. The simplest measure is the algebraic distance. A superellipse centred on the origin with axes aligned with the co-ordinate system can be represented by the following implicit equation:

$$\left(\frac{x}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\varepsilon}} = 1 \tag{8}$$

for which the algebraic distance is given by

$$Q(x,y) = \left(\frac{x}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\varepsilon}} - 1 \tag{9}$$

Again, this involves evaluating complex roots for negative values of x and y and choosing the root $x + j0$ and $y + j0$. Due to the symmetry of the superellipse this can be avoided by using the absolute values of x and y:

$$Q(x,y) = \left(\frac{|x|}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{|y|}{b}\right)^{\frac{2}{\varepsilon}} - 1 \tag{10}$$

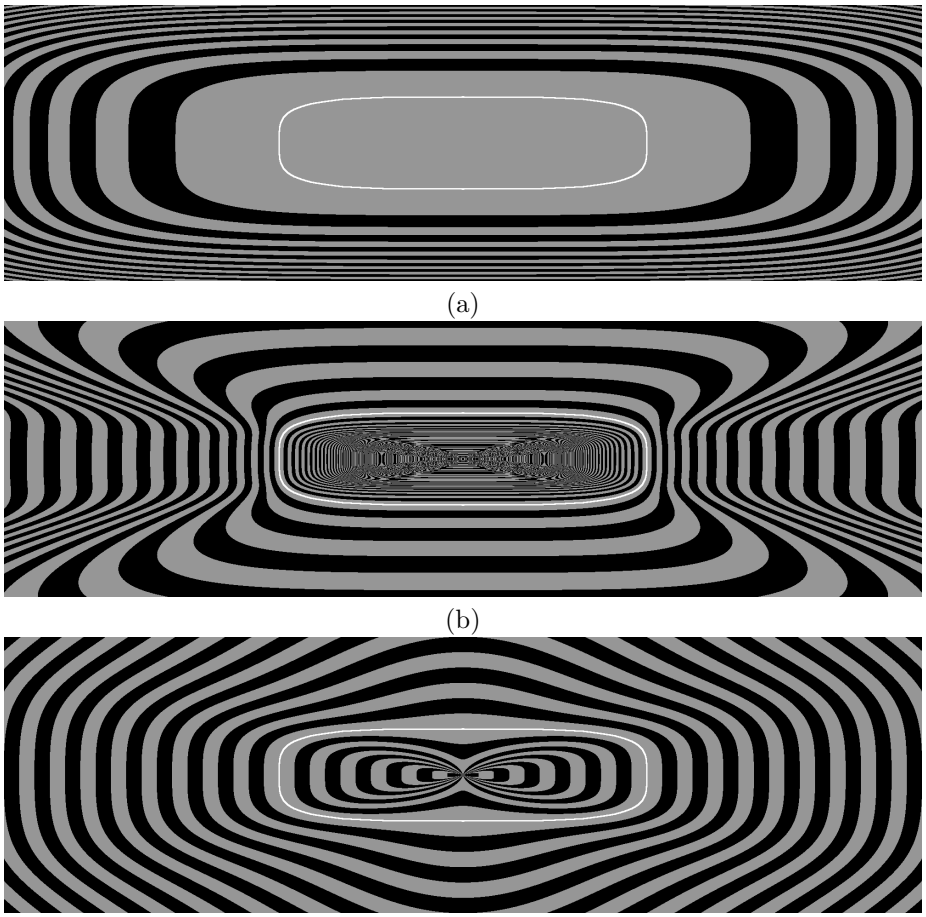
This would produce the curve in the positive x, positive y quadrant. The curve can then be reflected into the other three quadrants.

However, a problem with the algebraic distance is the so called *high curvature bias*. The algebraic distance from a point to the superellipse gives a lower estimate near high curvature sections than low curvature sections of the curve for the same Euclidean distance. When performing fitting this causes data points near the high curvature sections to have less influence than points near the low curvature sections since smaller errors are produced by the former points. This often results in the eccentricity of the fitted curves being overestimated. The curvature bias is evident in figure 3a which displays the error contours (alternatively coloured black and white) formed by the algebraic distance calculated from the superellipse boundary (the thin grey curve plotted at $Q(x,y) = 0$) for $a = 40$, $b = 100$ and $\varepsilon = 0.5$.

One popular method of approximately correcting the curvature bias is to inversely weight the algebraic distance by its gradient [1], thereby minimising the following:

$$\frac{Q(x,y)}{|\bigtriangledown Q(x,y)|} = \frac{\left(\frac{x}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\varepsilon}} - 1}{\sqrt{\left(\frac{2}{\varepsilon x}\left(\frac{x}{a}\right)^{\frac{2}{\varepsilon}}\right)^2 + \left(\frac{2}{\varepsilon y}\left(\frac{y}{b}\right)^{\frac{2}{\varepsilon}}\right)^2}} \tag{11}$$

However, the error contours for equation 11 displayed in figure 3b show there is still severe distortion in the error measure near the pointed ends of the superellipse. It can be seen that the values of the error within the superellipse are



(a)

(b)

(c)

Figure 3: Error contours for (a) algebraic distance to the superellipse, (b) algebraic distance divided the gradient of the algebraic distance, and (c) distance to the superellipse on a line through the centre of the superellipse.

significantly greater than values outside with the same distance to the boundary (shown by the closeness of the contours i.e. a high error gradient). This biases the fit to give greater significance to interior points. Moreover, we found that when minimising either equation 10 or 11 the system often incorrectly converged with very large values of ε. This arises since $\lim_{\varepsilon\to\infty}(|x|/a)^{2/\varepsilon} = 0$ when $|x|/a < 1$, and likewise for $(|y|/b)^{2/\varepsilon}$. The fitted superellipses tended to be very long as well as rectangular. When fitting superquadrics Solina and Bajcsy [21] suggested minimising

$$\sqrt{abc}\,Q(x, y, z) \tag{12}$$

so as to find the best fitting superquadric with the minimum volume. However, minimising a similar functional to fit superellipses (but ignoring the parameter c and dimension z in equation 12) proved to unstable, and was often minimised by setting $a = b = 0$.

Instead we choose to minimise the Euclidean distance d_p from the data point (x_p, y_p) to the point (x_s, y_s) on the superellipse along the line that passes through (x_p, y_p) and the centre of the superellipse $(0, 0)$, where:

$$\begin{aligned} d &= \sqrt{(x_p - x_s)^2 + (y_p - y_s)^2} \\ x_s &= \left| \frac{1}{\frac{1}{a^{\frac{2}{\varepsilon}}} + \left(\frac{y_p}{x_p b}\right)^{\frac{2}{\varepsilon}}} \right|^{\frac{\varepsilon}{2}} \\ y_s &= x_s \frac{y_p}{x_p} \end{aligned} \tag{13}$$

Figure 3c displays the error contours plotted for d_p. While still only approximating the shortest Euclidean distance from the point to the superellipse its distortions are less severe than either equation 10 or 11. In addition, equation 13 cannot be minimised simply by increasing the value of ε. The equivalent measure to equation 13 for superquadrics was also shown by Gross and Boult [9] to perform better than the preceding two.

The above equations have worked with object centred co-ordinates. To allow for translation by (x_c, y_c) and rotation θ equation 8 can be modified to:

$$\left(\frac{(x - x_c)\cos\theta - (y - y_c)\sin\theta}{a}\right)^{\frac{2}{\varepsilon}} + \left(\frac{(x - x_c)\sin\theta + (y - y_c)\cos\theta}{a}\right)^{\frac{2}{\varepsilon}} \tag{14}$$

which would result in the modification of equation 13. However, now that the superellipse is no longer symmetric in all four quadrants, complex roots cannot be avoided by simply using absolute values. This is avoided by keeping the superellipse centred at the origin and aligned with the co-ordinate axes. When fitting data, rather than transform the superellipse, the data is inversely transformed to fit the model.

Since the term being minimised is non-linear a closed form solution is not available. We use Powell's technique [16] which just requires the term being minimised to be evaluated. Powell's algorithm performs a gradient descent in the following way:

- Do gradient descent by varying the first parameter. Once the minimum has been found, repeat for the second parameter and so on for all parameters.
- Combine the changes in the parameters into a vector and minimise the function by gradient descent along this vector.
- Repeat from stage 1 until no change.

An advantage of this method is that the analytic form of the function is not required, the only requirement being the function must be able to evaluated at all required points. This considerably simplifies the minimisation procedure. Another advantage gained by not requiring the analytic form of the function is that alternative error can easily be implemented. Minimising the distance rather than squared distance reduced the effect of outliers. These can be further reduced by minimising not the summed error as above, but the median of the errors along the curve. This is the Least Median of Squares (LMedS) approach that has become popular in the field of robust statistics [20]. When using Powell's method of minimisation this adaptation is easily performed.

With iterative minimisation techniques it is important to provide a good initial estimate of the superellipse parameters. Initial work on fitting superquadrics (to range data) primarily dealt with unoccluded isolated objects [9, 14, 21]. This enabled the initial estimate to be determined by crude methods. For instance, Solina and Bajcsy used the smallest bounding box about the data to estimate the lengths of the three axes. More recently, some attention has been paid to occluded objects and cluttered scenes, e.g. Pentland [15]; Gupta and Bajcsy [10]; Yokoya *et al.* [26]; Dickinson *et al.* [6]; Leonardis *et al.* [12]. However, Pentland's technique requires searching over the entire range of possible superquadric occluding boundaries at different scales for potential concave silhouettes at all locations in the thresholded image; Gupta and Bajcsy's technique requires several parameters, reducing robustness; Yokoya *et al.* use simulated annealing, which is computationally expensive, and only show results for a simplistic scene containing one object; Dickinson *et al.* depend on the pre-segmentation stage being correct since there is no interaction between segmentation and fitting; and Leonardis *et al.* require a threshold specifying whether a model should grow.

Since we will generally only have a small portion of data underlying the superellipse the bounding box will not provide an adequate initial estimate. Instead we fit an ellipse to the data which provides reasonable estimates of all the parameters except for squareness which is initialised to 1.0. The ellipse is found by the closed form least square fitting of a conic [2]:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{15}$$

with the normalisation $F = 1$ to avoid the trivial solution $A = B = C = D = E = F = 0$. If the conic fit does not produce an ellipse then a circle fit is performed instead as the next best solution [23] and the orientation is estimated

by finding the principal axis of the data:

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{2\mu_{20} - 2\mu_{02}} \tag{16}$$

where μ_{pq} is the $(p,q)^{th}$ central moment calculated by

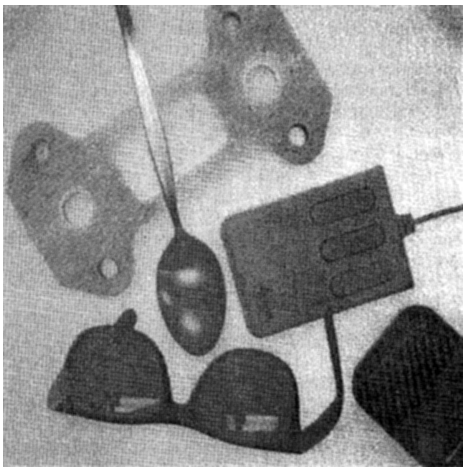
$$\mu_{pq} = \sum_i (x_i - x)^p (y_i - y)^q \tag{17}$$

where $(\bar{x}, \bar{y})$ is the centroid and there are i data points.

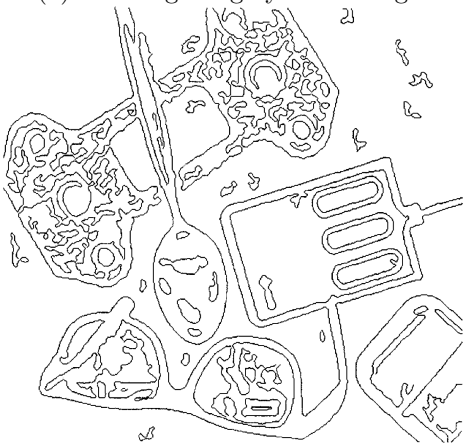
4 Results

Figure 4a shows an image containing a number of complex parts made up of straight lines, circular and elliptical arcs and freeform curves. The edgels extracted using the Marr-Hildreth edge detector with a $\sigma = 3.0$. This edge detector produces one pixel wide edge data resulting in the extraction of connected lists of edges using a simple edge following algorithm. Spurious short and weak edges are eliminated by using a threshold based on edge strength and edge length. The significant edges are shown in figure 4b. Figure 4c shows the results of the detection of superellipses. Crosses are used to indicate the ends of each superellipse. The lack of continuity between adjacent superellipses is mainly due to shortcomings of the plotting process. There are a number of interesting points to note about these results. The computer mouse has been segmented reasonably well. A single superellipse describes the inner edge of the mouse as a rectangular shape with rounded corners. The bumps on three sides of the edge have been removed from the segmentation tree since the overall shape of the mouse is considered more perceptually significant. The edges of each of the three buttons has been described by single superellipses. Note that instead of the boundary consisting of a rectangle with circular arcs at the ends (as would be expected from the edge data) the arcs are smaller and elliptical. As explained in section 5 a superellipse cannot represent such a shape precisely. Other interesting superellipses are the right hand corner of the large flat metallic object (just above the mouse buttons) which has been represented as a diamond ($\varepsilon = 1.0$) which adequately describes the vertex; rounded parts of the spectacles and spoon have been well represented; and the outer boundary of the object in the bottom right hand corner has been segmented into a number of straight lines and circular arcs. Examination of the superellipse parameters reveals that the straight lines are superellipses with low values for $\varepsilon \approx 0.1$ so that the diamonds have degenerated to crosses. The arcs are represented by an approximate circle and by a rounded square (see figure 5).

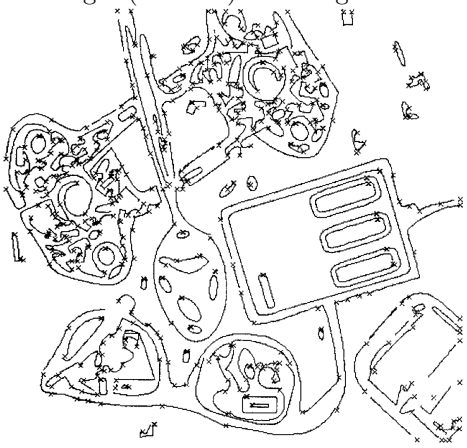
To further show the different superellipses a single connected edge list from figure 4b is isolated (figure 6a). Figure 6b shows the same data but with the full extent of the superellipses drawn in grey. The spoon bowl is approximated mainly by one ellipse and there are various combinations of rounded rectangles and diamonds for straight edges and corners.



(a) The original grey scale image



(b) The most significant edges (see text) after edge detection and thresholding.



(c) Superellipses detected with crosses delimiting each super ellipse.

Figure 4: Extracting superellipses from an image of objects containing straight, circular, elliptic and free-form curves.

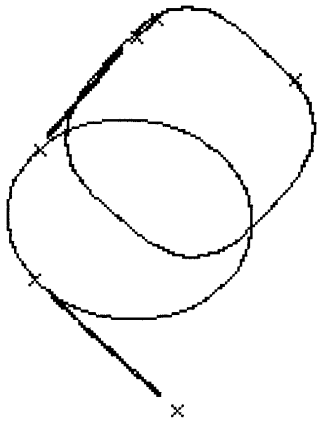


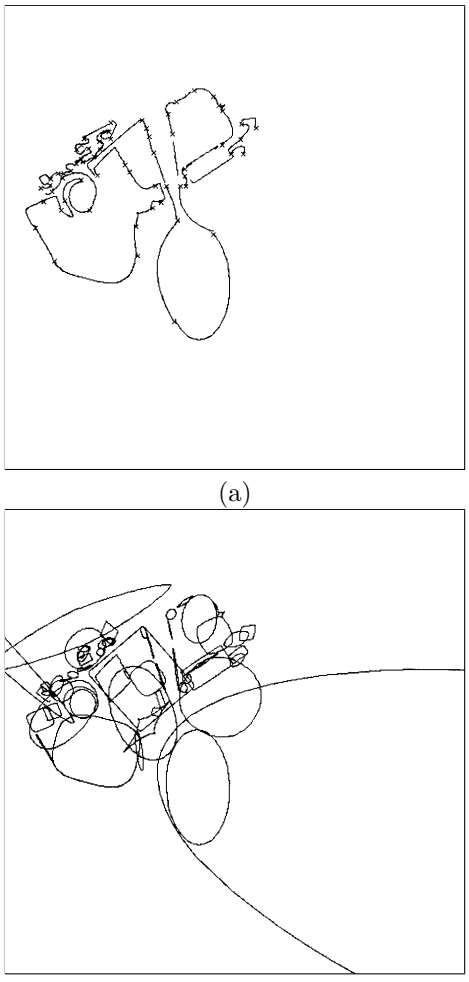
Figure 5: One list from figure 4c showing the full representation for each superellipse. One corner has been represented by an ellipse and the other by a diamond with rounded corners. The straight lines are represented by parts of diamonds, squares and crosses. Crosses show the ends of each superellipse.

As a final example consider the edge data of figure 7a that was obtained from a multiply exposed time lapsed photograph of a tennis player serving. The interesting feature here is the head of the racquet whose perceived shape changes throughout the serve. Figure 7b shows that in all cases a small number of superellipses represent the head of the racquet. In most cases a small number of superellipses (sometimes just one) represent the handle of the racquet.

This paper describes the extension of our previous algorithm which segmented curves into ellipses to superellipses which provide a more flexible description. It would also be possible to further extend the current proposal to yet more flexible representations such as hyperellipses, which have recently become popular [5, 11]. However, we feel that hyperellipses provide a less useful representation than superellipses. First, there are increased difficulties in estimating the parameters since: (i) the number of terms in the expression defining the hyperellipse has to be determined, and (ii) the increased number of free parameters increases the chance of poor fits arising from non-global minima. Second, the geometric interpretation of the parameters in terms of bounding lines does not explicitly describe the shape in the same was as the squareness parameter of the superellipse.

5 Discussion

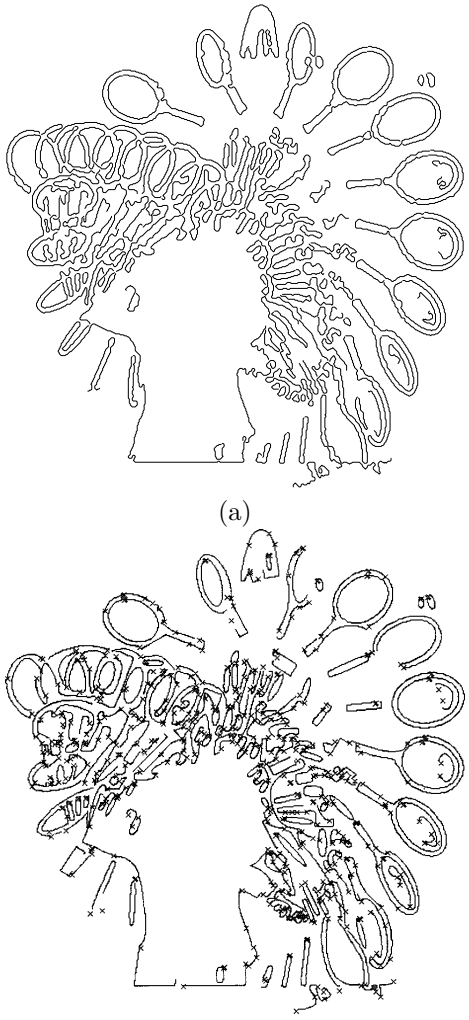
The results of segmenting image curves into superellipses give perceptually correct interpretations. The advantage of superellipses is that many different features can be represented by the same set of parameters such as ellipse, circles,



(a)

(b)

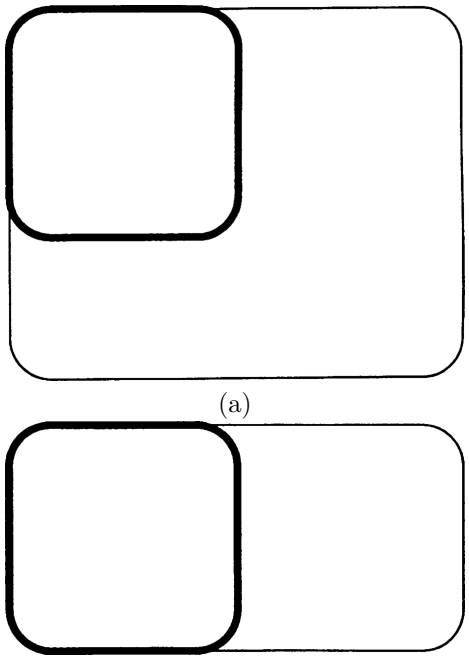
Figure 6: Super ellipse description of one long list of edge pixels for the spool and large flat object from figure 4c showing (a) the segmentation and (b) the full superellipses. Crosses delineate each superellipse



(a)

(b)

Figure 7: Segmentation of a multiple exposure image of a tennis player serving showing (a) the original edge data and (b) the superellipses. Each instance of the racquet and ball are clearly visible.



(a)

(b)

Figure 8: Examples for which a superellipse describing a rounded square (in bold) is the best fit to rectangular objects with circular rounded corners.

rectangles, corners and straight lines. A disadvantage is that different superellipses can represent the same curve data e.g. a 90° corner. There are at least three superellipse interpretations: (1) a diamond $\varepsilon = 2.0$, (2) a rectangle $\varepsilon \approx 0$, and (3) a pair of lines $\varepsilon = \{2, \infty, 0\}$. Note two lines will not be produced because the algorithm favours the longest interpretation. There is now an obvious ambiguity because it is difficult to decide on which of a rectangle or diamond is the best as they will both fit the data equally well in the sense of significance.

It may be necessary to consider segmenting the superellipses into other features such as rectangles with rounded corners into straight lines and circular arcs. Segmentation is possible in many cases by examining the parameters, especially ε which is the measure of squareness (see section 1.

A value of $\varepsilon = \{0, 2, \infty\}$ would indicate that the superellipse could be converted into a number of straight lines by locating the vertices. The segmentation points are determined by examining the parameters e.g. if $\varepsilon = 2$ then it is known that the vertices will be on the x and y axes of the superellipse i.e. $t = \{0, \pi/2, \pi, 3\pi/2\}$.

One important point concerns the description of rectangular shapes which have rounded corners. When considering an ellipse, the parameters a, b of equation 1 imply that a circle is scaled along one direction (compressed) to form an ellipse. For a rectangle, the same applies. Hence a square with rounded corners will have the corners represented by circular arcs. A rectangle is a square scaled along one axis which means the rounded corners are represented as elliptical arcs. It is therefore impossible to describe a rectangle with circular arcs as a single superellipse. An actual result (the mouse buttons) has been identified in section 4. The worst case would be a long rectangular object with rounded ends. This applies to complete figures. If only one or two corners of the figure are visible it can be perfectly modelled by a superellipse with axes of equal length giving a rounded square which will have circular arcs at the corners. Figure 8 shows examples.

It is important to note the non-linear sensitivity of the shape of the superellipse to values of ε. The shape of the superellipse is most sensitive for $\varepsilon \approx 1.0$ (an ellipse). Varying ε from 0.1 to 20.0 gives almost all shapes possible. Outside these values there is little difference in shape, especially in the context of images of less than 512×512 pixels in size.

Segmentation of a superellipse into subsections requires the formulation of curvature κ as a function of t. Considering equation 1 the derivatives of x and y w.r.t. t are:

$$\begin{aligned} \dot{x} &= -a\varepsilon \cos^{\varepsilon-1} t \sin t \\ \dot{y} &= b\varepsilon \sin^{\varepsilon-1} t \cos t \end{aligned} \tag{18}$$

and the second derivatives are:

$$\begin{aligned} \ddot{x} &= a\varepsilon^2 \cos^{\varepsilon-2} t - a\varepsilon^2 \cos^{\varepsilon} t - a\varepsilon \cos^{\varepsilon-2} t \\ \ddot{y} &= b\varepsilon \sin^{\varepsilon-2} t \cos^2 t(\varepsilon - 1) - b\varepsilon \sin^{\varepsilon} t \end{aligned} \tag{19}$$

The curvature can be expressed as:

$$\kappa = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x} + \dot{y})^{3/2}} \tag{20}$$

By substituting the values from equations 18 and 19 into equation 20, an expression for κ w.r.t. t can be determined. Given some knowledge of the squareness ε the superellipse can be segmented into regions of high curvature (corners) and low curvature (approximate straight lines).

The current version of the algorithm using Powell's technique requires many hours of computation on a Sun IPC workstation for an image such as in figure 4a. Various approaches are being investigated to speed up the algorithm. For instance, minimisation techniques such as the Levenberg-Marquardt technique [16] which use gradient information are faster than Powell's method. However, the partial derivatives of equation 13 are relatively complex (see Appendix). Currently at each level in the tree search all the edge data is used to determine the fit. The use of a multi-resolution technique which starts with a small subset of points for the initial fit and successively fits to more and more data would improve performance [21]. Early jump-out methods could be used [3] but these need some threshold which we wish to avoid.

6 Conclusions

This paper has described an algorithm that segments curve data into a number of superellipses. The results indicate that it is possible to segment image curves in a bottom-up process (very little contextual information) into perceptually significant high order representations. The use of a significance measure enables a perceptually meaningful description to be determined as opposed to a minimum error description. There is an increase in complexity of the algorithm compared to our previous work involving segmenting curves into straight lines, arcs and ellipses, The squareness parameter introduces non-linearities, requiring an iterative method for minimising the error of fit between the superellipse and the data.

Section 5 discussed the segmentation of superellipses into other features such as lines and arcs. To make the algorithm more general purpose, extra parameters could be incorporated to describe deformations such as tapering and bending [21]. Further details of the segmentation technique described in this paper are available [24].

References

[1] G. J. Agin. Fitting ellipses and general second order curves. Technical Report CMU-RI-TR-81-5, Robotics Institute, Carnegie-Mellon University, USA, 1981.

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

[3] D. I. Barnea and H. F. Silverman. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, 21(2):179–186, 1972.

[4] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics Applications*, 1:11–23, 1981.

[5] I. Cohen and L.D. Cohen. A hyperquadric model for 2-d and 3-d data fitting. In *Int. Conf. Patt. Recog.*, volume B, pages 403–405, 1994.

[6] S.J. Dickinson, D. Metaxas, and A. Pentland. Constrained recovery of deformable models from range data. In C. Arcelli *et al.*, editor, *Aspects of Visual Form Processing*, pages 158–167. World Scientific, 1994.

[7] R.O. Duda and P.E. Hart. *Pattern Recognition and Scene Analysis*. Wiley, 1973.

[8] M. Gardiner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, 21:222–234, 1965.

[9] A. D. Gross and T. E. Boult. Error of fit for recovering parametric solids. In *2nd. International Conference on Computer Vision, Tampa, Florida, USA*, pages 690–694. IEEE, 1988.

[10] A. Gupta and R. Bajcsy. Surface and volumetric segmentation of range images using biquadrics and superquadrics. In *International Conference on Pattern Recognition, The Hague, Netherlands*, pages 158–162. IEEE, 1992.

[11] S. Han, D.B. Goldgof, , and K.W. Bowyer. A hyperquadric model for 2-d and 3-d data fitting. In *Int. Conf. Comp. Vision*, pages 492–496, 1993.

[12] A. Leonardis, F. Solin, and A. Macerl. A direct recovery of superquadric models in range images using recover-and-select paradigm. In *Europ. Conf. Comp. Vision*, volume A, pages 309–318, 1994.

[13] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[14] A. Pentland. Perceptual organisation and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[15] A. P. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4:107–126, 1990.

[16] W. H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C*. Cambridge University Press, Cambridge, UK, 2nd. edition, 1992.

[17] P. L. Rosin. A note on the least squares fitting of ellipses. *Pattern Recognition Letters*, 14, 1993.

[18] P. L. Rosin. Non-parametric multiscale curve smoothing. *Int. J. of Pattern Recognition and Artificial Intelligence*, 8:1381–1406, 1994.

[19] P. L. Rosin and G. A. W. West. Non-parametric segmentation of curves into various representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted for publication.

[20] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection.* Wiley, 1987.

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

[22] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.

[23] S. M. Thomas and Y. T. Chan. A simple approach for the estimation of circular arc centre and its radius. *Computer Vision, Graphics and Image Processing*, 45:362–370, 1989.

[24] G. A. W. West and P. L. Rosin. Using superellipses to represent 2D curves. Technical Report 93/23, Department of Computer Science, Curtin University, Perth, WA, Australia, 1993.

[25] G. A. W. West and P.L. Rosin. Techniques for segmenting image curves into meaningful descriptions. *Pattern Recognition*, 24:643–652, 1991.

[26] N. Yokoya, M. Kaneta, and K. Yamamoto. Recovery of superquadric primitives from a range image using simulated annealing. In *International Conference on Pattern Recognition, The Hague, Netherlands*, pages 168–172. IEEE, 1992.

Appendix

The partial derivatives of the distance measure given in equation 13 were found using Maple and (after some tidying up) are:

$$\frac{\delta d}{\delta a} = \frac{-\sqrt{x_p^2+y_p^2}\left(1+y_p^{\frac{2}{\epsilon}}x_p^{-\frac{2}{\epsilon}}b^{-\frac{2}{\epsilon}}a^{\frac{2}{\epsilon}}\right)^{-\epsilon}b^{\frac{2}{\epsilon}}x_p^{\frac{2}{\epsilon}}\left(x_p\frac{t^{\frac{\epsilon}{2}}}{x_pb}-a\right)}{\sqrt{x_p}\left(x_p^{\frac{2}{\epsilon}}b^{\frac{2}{\epsilon}}+y_p^{\frac{2}{\epsilon}}a^{\frac{2}{\epsilon}}\right)\sqrt{x_pt^{\epsilon}-2x_pbat^{\frac{\epsilon}{2}}+b^2a^2x_p}} \tag{21}$$

$$\frac{\delta d}{\delta b} = \frac{\sqrt{x_p^2+y_p^2}y_p^{\frac{2}{\epsilon}}a^{1+\frac{2}{\epsilon}}\left(x_p\sqrt{u}-a\right)}{{x_p}^2btu\sqrt{\left(1-2bat^{\frac{-\epsilon}{2}}+t^{-\epsilon}a^2b^2\right)}} \tag{22}$$

$$\frac{\delta d}{\delta \epsilon} = \frac{\left(x_p^2+y_p^2\right){x_p}^{\frac{2}{\epsilon}+\frac{1}{2}}t^{\frac{\epsilon}{2}}ab^2\left(x_pt^{\epsilon+1}-a\right)}{\left(2\epsilon t^{\epsilon+1}\sqrt{\left(x_pt^{\epsilon}-2x_pbat^{\frac{\epsilon}{2}}+b^2a^2x_p\right)\left(x_p^2+y_p^2\right)}\right)}\times \left(vb^{\frac{2}{\epsilon}}+(v+2)\left(\frac{ay_p}{x_p}\right)^{\frac{2}{\epsilon}}\left(\ln(x_p)+\ln(b)-\ln(a)-\ln(y_p)\right)\right) \tag{23}$$

where

$$t = {x_p}^{\frac{2}{\epsilon}}b^{\frac{2}{\epsilon}}+{y_p}^{\frac{2}{\epsilon}}a^{\frac{2}{\epsilon}},\quad u=\frac{t^{\epsilon}}{{x_p}^2b^2},\quad v=\epsilon\ln\left(1+\left(\frac{y_pa}{x_pb}\right)^{\frac{2}{\epsilon}}\right)$$