Multistage Combined Ellipse and Line Detection

Geoff A.W. West & Paul L. Rosin
Cognitive Systems Group
School of Computing Science Curtin University of Technology
GPO U1987, Perth, 6000, Western Australia
email: geoff@hinault.cs.curtin.edu.au rosin@marsh.cs.curtin.edu.au

Abstract

This paper describes an algorithm for the detection of ellipses and lines in image edge data. Connected edge pixels are transformed into polygonal approximations by a two stage algorithm. Then a second two stage algorithm replaces combinations of lines by ellipses if the ellipse fit is better. For each algorithm a combination of splitting and merging of the data is used to enable global and local constraints to fit different representations to the pixel data. All merging decisions use a significance measure to replace a number of representations by a single representation which removes the need for thresholds in the algorithms. The structure of the algorithm allows any particular representation to describe the data e.g. parabolae, splines, etc. instead of ellipses.

1 Introduction

In computer vision the extraction of meaningful features from images is an important technigue. The most popular approach is based on edge detection. For model based object recognition, edges must be represented in a more manageable form than simply pixels. The type of representation required is highly application dependent but is typically based on a combination of straight line approximations and higher order curves such as arcs, conic sections, spline and curvature primitives.

A number of techniques have been proposed for determining polygonal approximations [5] [10]. However it is only recently that attention has been concentrated on the extraction of higher order representations because of the increased number of parameters or degrees of freedom and the ill-conditioned nature of the problem [8] [4].

This paper describes a technique for generating a higher order description by segmenting the edge data into combinations of ellipses and lines. There are four stages used: (1) lines are fitted to connected lists of edge pixels using Lowe's technique [3], (2) lines are grown by combining adjacent lines, (3) ellipses replace lines and finally (4) ellipses are grown by combining with adjacent lines and/or ellipses. In all stages replacement occurs only if the resultant fit is better. The concept of better fit is that suggested by Lowe which is termed a measure of significance. Significance is the maximum error between the fitted representation and the data (a line fitted to pixels or an ellipse fitted to lines) normalised by the length of the representation. The significance is a

BMVC 1992 doi:10.5244/C.6.21

scale invariant measure which allows the replacing of (i) pixels by a line, (ii) combinations of lines by a line, (iii) combinations of lines by ellipses and (iv) combinations of lines and ellipses by ellipses. The same measure of significance is used in all four stages removing the requirement for any thresholds.

Previously published results [8] have demonstrated the utility of ellipse and line detection based on stages (1) and (3) of the technique described above. In this paper an improved version of the line and ellipse fitting algorithms is described which overcomes the disadvantages of using a binary search tree for stages (1) and (3) by adding new stages (2) and (4). It is shown that the addition of these stages improves the performance of the algorithm. In addition the improvements have been added to the algorithm for detecting arcs and lines (the LAD algorithm [11]).

2 Finding Representations

The problem of finding the optimum segmentation and hence representation for any curve has resulted in a number of proposed solutions. Simple local techniques such as segmenting at points of high curvature (vertices) and points of change in the rate of change of curvature have been proposed as have more global techniques such as segmenting at the point of maximum distance from the curve to the representation. These usually result in a sub-optimal result. To overcome the problem of using local constraints, context dependent local and global techniques have been proposed. Fischler [2] investigated the performance of people for segmenting curves for a number of objectives and discovered that the points of segmentation varied depending on the objective. From the results he proposed a technique that processed curves using large windows to attempt to capture more global information. In fact it is only possible to correctly segment a curve by taking into account the process that formed the curve e.g. the resulting 2D projections of 3D objects under certain viewing conditions. In addition a large number of curve models need to be available e.g. sine waves, parabolas, etc.

Where the objective is to fit a particular representation, an almost optimal technique can be formulated. All possible combinations of a particular representation such as a line can be fitted to the curve and the combination with the best goodness of fit chosen. Consider the case of dividing a curve up into n segments of equal numbers of pixels. The task is to determine which of the possible segmentation points can be removed, e.g. by combining two adjacent segments to form one line. This is a combinatorially expensive process of order $O(2^{n-1})$. For example, for a curve 40 pixels long where n=8, eight 5 pixel long lines is the maximum number of lines that can fit the curve and the minimum is one. To determine the best combination, 128 combinations need to be considered. The combinatorics are compounded if an attempt is made to replace combinations of lines by ellipses or some other representation in a second stage of processing, or if different length segments are considered.

It would appear the best technique would be to restrict the number of combinations that need to be tested. This can be achieved by using a binary search tree as proposed by many researchers [3] [6]. However this has disadvantages as shown in section 3 because some combinations of adjacent representations are never compared as they are in different branches of the tree. To improve

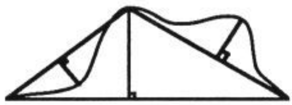

Figure 1. Segmentation of curves

the results a second stage is added that compares these representations.

There are a number of other issues that have to be addressed when considering the segmentation of edge data into higher order representations. The majority of algorithms, with the exception of [1], depend on pre-set parameters to determine such things as the accuracy of fit, the scale at which breakpoints are located, where breakpoints are, and thresholds for selecting the breakpoints. However, the results obtained are dependent on the parameters. For the line and ellipse fitting algorithm [8], breakpoints for the two stages are the well known points of maximum error between the fitted straight line and the data. These have the advantage of not requiring any parameters for their detection. A subset of these become the vertices in the polygonal representation resulting from the first stage. In the second stage groups of lines are replaced by ellipses so the breakpoints between representations are still effectively vertices. However, these may not be optimal breakpoints for higher order representations such as ellipses and it can be argued that other breakpoints such as points of inflection should also be used. Most higher order representations have continuously varying curvature so the detection of breakpoints based on changes in curvature is not valid. Higher order differentials are necessary which are difficult to determine in discrete data.

3 Line Fitting

3.1 Stage (1): binary search tree

The line fitting technique used is the familiar recursive binary tree search. Each curve is hypothesised to be a straight line, figure 1, and segmented at the point of maximum deviation from the curve to the straight line. The process is then repeated for each of the two curves recursively. When the bottom of the tree is reached and the curve cannot be subdivided anymore, tail recursion is used to combine together those lines for which the combined line is a better representation than the other lines. This algorithm has been described in detail elsewhere [11]. The important points to note are (1) the use of points of maximum deviation for breakpoints and (2) the binary search tree preventing all adjacent combinations of lines from being compared. Consider the curve of figure 2a, the result of the algorithm is shown in 2b whereas the intuitively correct result is that in 2c. Figure 3 shows the interpretation tree for this curve. The tree shows that the curve has been segmented into the lines a,b,c,d,e and f, some of which have been combined under tail recursion to give the result of figure 2b.

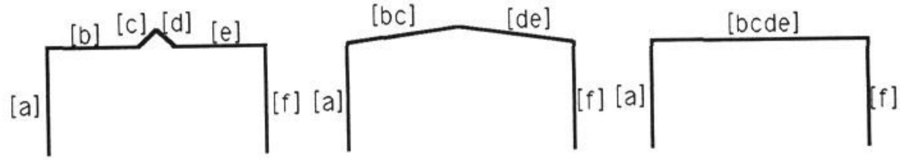

Figure 2. (a) original curve, (b) incorrect interpretation and (c) correct interpretation

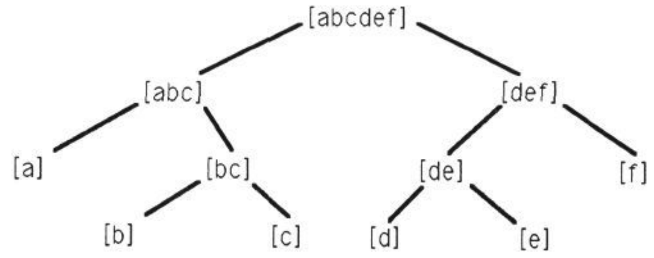

Figure 3. The interpretation tree for figure 2

3.2 Stage (2): combining adjacent lines

To overcome deficiencies such as shown in figure 2 a second stage is used that compares all adjacent lines and combines them if possible. Consider the example of figure 1 again. Figure 3 shows the tree that results from processing this curve. Note the two lines that replace parts of the original curve [bc] and [de] are not compared at all in the tree, hence the incorrect result. The second stage takes the representations [a], [bc], [de] and [f] and computes the goodness of fit for combinations of a selected representation and its neighbours. Possible representations centred on [bc] when compared and possibly combined with its neighbours are: [abcde], [abc]&[de], [a]&[bcde], and [a]&[bc]&[de] (unchanged) where [AB] indicates one representation made up of previous representations [A] and [B]. Note that [abc] has already been tested as [a] and [bc] are adjacent in the tree. The new representation is the one that gives the lowest significance, the same measure used for the first stage. By iterating over the whole curve until no further improvements can be made, all adjacent combinations are tested. Using the second stage on the example of figure 2b should result in representations [bc] and [de] being combined resulting in the result of figure 2c - the intuitively correct result. This is because the combination of [bc] and [de] should result in a lower significance than [bc] and [de] individually.

For part of the curve of figure 2b, segment [bc] and [de] each have a significance of approximately 1/6 and segment [bcde] has a significance of 1/12. Hence [bcde] is a better representation that [bc] or [de]. For a number of real images, table 1 shows results for the total number of lines that result from the original and improved line fitting algorithm. Each of these images contain a number of generalised cylinders. *iccv32* is the image for which results have previously been presented for line and ellipse fitting [8].

	Input data	After 1st stage	After 2nd stage
Image name	Number of pixels	Number of lines	Number of lines
mugs6	20870	1195	1109
iccv32	14391	897	728
can3	12887	917	774
cyl21	17659	1257	1166

Table 1. Results for line fitting.

Stage (2) has improved the results in all three images reducing the total number of lines. The use of significance means that the algorithm favours lines that are more significant than others. Note a weighting factor can be used to force the algorithm to favour new combinations i.e. longer lines. Weighting the new combinations by a factor less than 1.0 will give the new line an artificially lower significance forcing the replacement by a longer line.

Figure 4 shows results for the image *cyl21*. For the original image of figure 4a, figure 4b shows the results of the new algorithm. The differences between the results of the old and new algorithms are shown in figures 4c and 4d. Figure 4d shows the lines that replace the lines shown in figure 4c. Note that many old lines have been replaced by fewer longer new lines.

4 Ellipse Fitting

4.1 Stage(3): binary search tree

The basic ellipse fitting algorithm has been presented elsewhere [8]. However a short description is required for completeness. The result of the line fitting algorithm is used as the input of the ellipse fitting algorithm which is, like the line fitting algorithm, based on a binary search. Figure 5a shows the output of line fitting for a curve consisting of two straight lines and one ellipse. In the original algorithm, an iterative Kalman filter was used to fit an ellipse to the endpoints of the curve. A standard least mean square conic fitting algorithm is now used. Although this can generate hyperbolae and parabolae, it usually generates ellipses if the curve is an ellipse. Non-ellipse fits are ignored by giving the result a high significance value so it is always replaced. Replacing the Kalman filter has not altered the results significantly but the algorithm is now faster. Then the list of lines is split into two lists at the vertex of maximum deviation from the ellipse. The algorithm is then repeated for the two lists. Figure 5b shows the resulting interpretation after ellipse fitting which is not correct. The single elliptic arc has been segmented into two elliptic arcs. The intuitively correct result is shown in figure 5c. Figure 6 shows the interpretation tree for figure 5b. Note that the two elliptic arcs [bcd] and [efg] do not get compared in the tree.

When the length of a list of lines reduces to only 5 lines (or 6 vertices) the ellipse fitting is halted and tail recursion used to choose the combination of lines and ellipses that best describe the list of lines in the sense of significance. At each node in the tree the single ellipse describing the list of lines is chosen if it has a better significance than any of the representations below it in the tree.

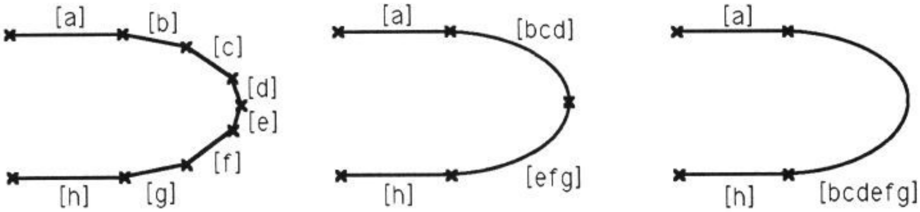

Figure 5. (a) result of line fitting, (b) incorrect oversegmentation, (c) correct segmentation

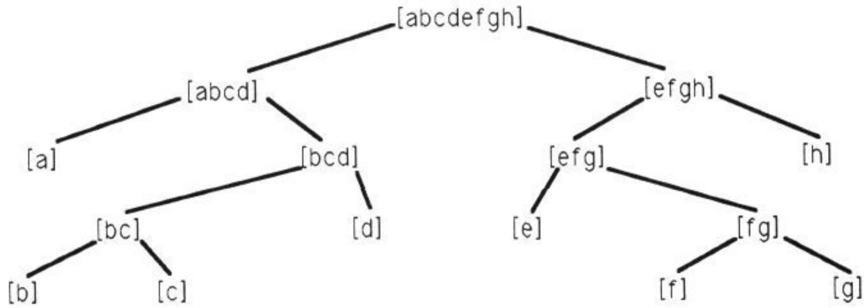

Figure 6. Interpretation tree for figure 5

A number of limitations occur with the algorithm which can be minimised by the following techniques.

4.2 The use of soft breakpoints

Fitting ellipses to line data is more efficient that fitting to pixel data because of the reduction in the number of points for ellipse fitting. However this prevents attempts to fit an ellipse to less that 5 lines (6 vertices) since the fit will be underdetermined. This leads to poor results as ellipses may be missed. It may be that an ellipse would be a better fit to the pixel data than the line. To overcome the problem of not being able to fit an ellipse to less that 5 lines, the concept of soft breakpoints (edge pixel coordinates) is introduced. Consider the hypothesis that an ellipse fits the pixel data better than one line. To confirm the hypothesis 6 data points are required so 4 soft breaks points taken equidistant along the pixel data are used along with the two vertices of the line. If the ellipse is a good fit to the data, a lower significance value should occur. This is the worst case scenario and the algorithm is adaptive such that soft break points are only used when less that 5 lines are used. With three lines only four vertices are available so one additional soft break point is needed for each line. For two lines, two soft breakpoints per line are required. As such it a variable resolution technique that adapts to the amount of data present using the minimum required for the fitting so reducing computation. The alternative of always fitting an ellipse directly to the pixel data is rejected because of the increased computation in the least mean square conic fitting.

4.3 Stage (4): combining ellipses with adjacent ellipses/lines

To overcome the problem of using a binary tree, combinations of adjacent ellipses and lines are tested in a way similar to that described for stage (2) of line fitting. An ellipse will be combined with one or both of its neighbouring lines/ellipses if the significance is better.

Stage (4) is an iterative process that examines each ellipse and determines if it can be extended by combining it with the adjacent representation (line or ellipse). There are four possible outcomes: (i) do not combine, (ii) combine with the left representation, (iii) combine with the right representation and (iv) combine with both representations. For each of these possibilities, the significance measure (as used in the previous three stages) is determined. The resulting representation chosen is the one that results in lower significances (and hence more accurate fits). Iterating over all the ellipses until no further changes can be made results in ellipses growing until further increase in size results in reduced accuracy of fit.

	Input data	After stage 3		After stage 4	
Image name	No. of lines	No. of lines	No. of ellipses	No. of lines	No. of ellipses
mugs6	1108	489	132	447	130
iccv32	728	393	76	381	72
can3	774	451	89	432	88
cyl21	1166	664	142	620	140

Table 2. Results for ellipse fitting.

The results of table 2 show that the additional stage of ellipse growing results in reduced numbers of ellipses and straight lines. Longer ellipses are being generated by combining with lines and with some other ellipses. It is interesting to note that there is an appreciable reduction in the number of lines which implies the use of the point of maximum deviation is not an optimal segmentation method for ellipses. However the shortcomings are reduced by the use of the new stage. The effect of the new algorithm is shown in figure 7 for image *cyl21*. Figure 7a shows the results of the improved algorithm. Figures 7b and 7c show the differences between the two algorithms. Note that in most cases one ellipse has been grown by combining it with lines. However, in some cases, an ellipse has been grown by combining with ellipses and lines.

5 Results

To further demonstrate the performance of the new algorithm, results for a number of images are shown in table 3. The results of the new algorithm are shown along with the differences between the new four stage algorithm and the original two stage algorithm. The differences show the original combinations of features and those replaced by the new stages.

	Two stage algorithm			Four stage algorithm		
	After stage 1	After stage 2	After stage 2	After stages 1 and 2	After stages 3 and 4	After stages 3 and 4
Image name	No. of lines	No. of lines	No. of ellipses	No. of lines	No. of lines	No. of ellipses
mugs6	1195	597	118	1109	447	130
cyl21	1257	924	77	1166	620	140
iccv32	897	550	56	728	381	72
can3	917	626	48	774	432	88
cup3	518	358	33	477	270	53

Table 3. Results for the old two stage algorithm and the new four stage algorithm.

For all the images, the number of lines has been reduced and the number of ellipses has increased. This is expected because more ellipses are being detected at the expense of lines. Figure 8 shows some of a sequence of images of a predominantly circular object. The straight lines and ellipses have been extracted in all the images and can be tracked reasonably well.

6 Conclusion

The new four stage algorithm shows a significant improvement in performance over the original two stage algorithm [8] for a small increase in computation. In the spirit of the original algorithm, the new algorithm avoids the need for any parameters. There are two reasons for the improvement. The first reason is the use of soft breakpoints meaning that ellipses are adaptively fitted to what can be regarded as the minimum number of points. For a curve consisting of a large number of lines, just the vertices are used. For a part of the curve consisting of less than 5 lines, other points are used to allow an ellipse to be fitted in the overdetermined sense. The effect of soft breakpoints is to enable ellipses to be fitted to more parts of the data than before resulting in more ellipses being detected. The second reason is the use of the extra stages to overcome the shortcomings of using a binary search tree. Lines and ellipses already detected can be extended by combining with other ellipses and/or lines. Currently a minor extension to deal with closed boundaries is being carried out. The combining of adjacent lines/ellipses needs to be performed between the first and last elements of a closed list. Although this is a minor extension it is necessary for completeness. A further extension which can be used is to refit the hypothesised ellipses to the original pixel data to get a better estimate of each ellipse. Hence a good estimate of each ellipse can be determined after segmentation.

The algorithm is made up of a number of stages that can be regarded as splits and merges. The sequence of operations is split and merge (stage 1), merge (stage 2), split and merge (stage 3) and merge (stage 4). Each split uses global information to determine breakpoints and each merge uses local information to remove breakpoints.

The improvements increase the usefulness of the algorithm for many applications which require the detection of ellipses in the 2D image (circles in 3D space) such as detection of generalised cylinders [9] and for bottom up perceptual grouping [7] [12].

Finally the algorithm can be easily modified such that any parametric curve can be detected if a fitting algorithm is available e.g. splines and parabolae, and has been extended to 3D curve data [13].

References

[1] C.-H.Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. PAMI*, 11:859–872, 1989.

[2] M.A. Fischler and R.C. Bolles. Perceptual organisation and curve partitioning. *Trans. IEEE PAMI*, 8:100–105, 1986.

[3] D.L. Lowe. Three-dimensional object recognition from single two-dimensional views. *AI*, 31:355–395, 1987.

[4] T. Pavlidis. Curve fitting with conic splines. *ACM Trans. Graphics*, 2:1–32, 1983.

[5] T. Pavlidis and S.L. Horowitz. Segmentation of plane curves. *IEEE Trans. on Computers*, 23:860–870, 1974.

[6] P.L. Rosin and G.A.W. West. Segmentation of edges into lines and arcs. *IVC*, 7:109–114, 1989.

[7] P.L. Rosin and G.A.W. West. Perceptual grouping of circular arcs under projection. *Proc. 1st British Machine Vision Conf.*, pages 379–382, 1990.

[8] P.L. Rosin and G.A.W. West. Segmenting curves into elliptic arcs and straight lines. *Proc. IEEE 3rd Int. Conf. on Comp. Vision*, pages 75–78, 1990.

[9] P.L. Rosin and G.A.W. West. Extracting surfaces of revolution by perceptual grouping of ellipses. *Proc. IEEE Comp. Soc. Conf. on Comp. Vision and Pat. Rec.*, pages 677–678, 1991.

[10] Ramer U. An iterative procedure for the polygonal approximation of plane curves. *CGIP*, 1:244–256, 1972.

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

[12] G.A.W. West and P.L. Rosin. Non-parametric segmentation of 2d and 3d curves into lines and arcs. *Proc. 2nd Int. Conf. of Automation, Robotics and Computer Vision - ICARV92*, 1992.

[13] G.A.W. West and P.L. Rosin. Perceptual grouping of parallel planar ellipses for the recognition of 3d objects. *Proc. 2nd Int. Conf. on Image Processing - ICIP92*, 1992.

Figure 4 (a) Raytraced image of cylinders.

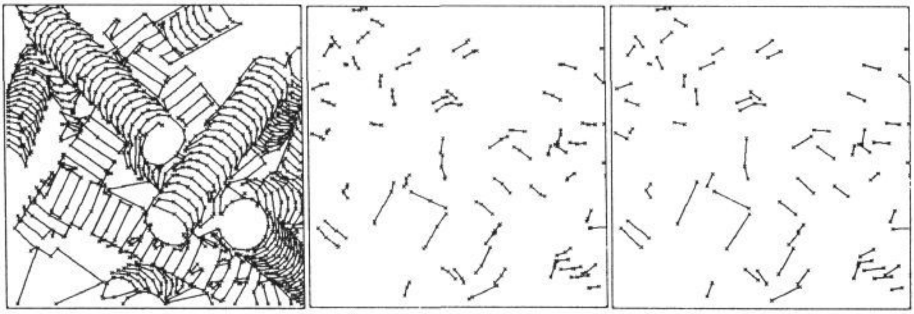

Figure 4 (b) Lines detected by improved algorithm.
Old lines (c) replaced by new algorithm with single lines (d)

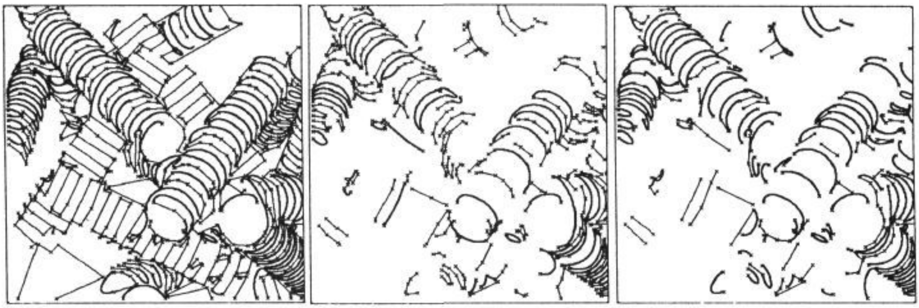

Figure 7 (a) Ellipses and lines detected by improved algorithm
Old lines and ellipses (b) replaced by single ellipses (c).

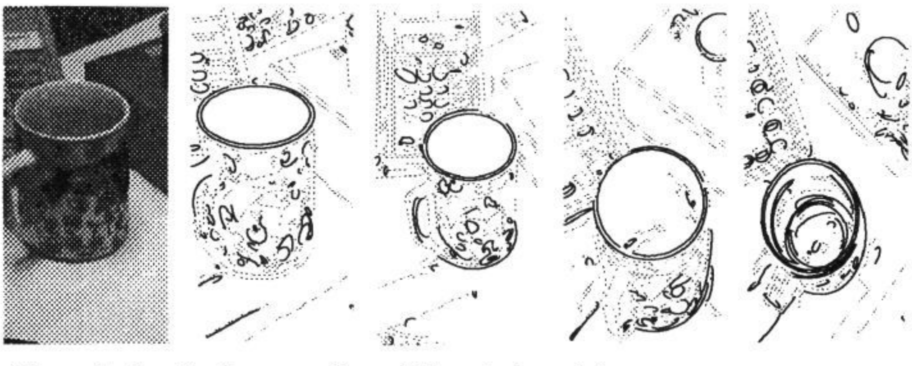

Figure 8. Results for scene from different viewpoints.