

Grouping Curved Lines

Paul L. Rosin
Institute for Remote Sensing Applications
Joint Research Centre
21020 Ispra (VA), Italy
email: paul.rosin@jrc.it

Abstract

This paper examines the problem of automatically grouping image curves. In contrast, most previous work has been restricted to points and straight lines. Some of the computational aspects of the groupings of continuation, parallelism, and proximity are analysed, and the issues of neighbourhoods, combinatorix, and multiple scales are discussed.

1 Introduction

A common process, both in human and computer vision, is the grouping of features extracted from the scene. The majority of previous techniques developed for grouping have been based on dot patterns [1] or straight lines [7], although there has been some recent interest in ellipse grouping [15] and curve grouping [3, 11, 16]. Rules for grouping curves need to be more complex than rules for grouping simpler features since arbitrary curves have more degrees of freedom than dots, straight lines, etc.

Early this century the Gestalt school of psychology studied the grouping of sensory phenomena, and identified a number of classes of grouping: proximity, similarity, closure, good continuation, and symmetry. Research in computer vision has usually restricted itself to using one or several of these types of groupings. In this paper we shall concentrate on curvilinearity (good continuation), parallelism, and proximity.

2 Issues in automating curve grouping

We first discuss several general issues that apply to all the types of grouping operations: the appropriate scale(s) of analysis; methods for segmenting and completing curves; and the selection of which subsets of curves are candidates for grouping together.

2.1 Multi-Scale vs. Natural Scale vs. Single Scale

Since real image curves contain noise, irrelevant detail, and differently sized structures, it is necessary to take scale into account. When tackling the problem of scale there are three main approaches that can be taken. The most straight forward is to represent the curves over a fixed range of scales [8]. Although this is

a robust approach it produces a vast amount of data. A second approach is to represent each curve at only a small number of interesting or significant scales which we have called their natural scales [13]. Between each natural scale there is some qualitative change, e.g. they contain different structures. Finally, the curve can be represented by a single smoothed version. Either the whole curve is smoothed uniformly at a single scale or by different amounts over individual sections [14]. There is a trade-off between completeness and conciseness that is balanced differently by the three approaches.

2.2 Segmentation and Completion

In this paper we only consider bottom-up grouping, and so extending a curve is a local operation (although see Elder & Zucker [4] who argue that completion is a global process). Rather than determine the extension from just the endpoints and their tangents we use a section of the curve delimited by the endpoint to the first singular point (i.e. either a curvature maximum or minimum, or a zero-crossing of curvature). Singular points are commonly used to represent curves since they are the most salient locations [5]. Moreover, their position varies little (and is invariant in the case of zero-crossings) under perspective projection. A similar approach is taken by Mohan & Nevatia [11] who segment curves at extrema of curvature prior to performing grouping. Another reason for segmentation is that unlike points or straight lines, some curves can be grouped with themselves.

The gap between two curves is interpolated by fitting a smooth curve through the pairs of endpoints and singular points. Many techniques have been suggested, although most give similar looking results. We do not consider the exact shape of the curve to be crucial, and use a simple technique [10] which fits a smooth curve such that the curvature at a knot point is equal to that of the circle through the knot and its two adjacent knots. A curve can also be extrapolated by the same method using the endpoint and three singular points.

2.3 Grouping Selection

Although it is possible to consider all pairwise (or higher-order) combinations of curves for grouping, this leads to a proliferation of groupings. Instead, it is common to select pairs of candidate curves which are suitable for grouping together by applying a distance threshold, discarding pairs of curves separated beyond this distance. The threshold is some factor (usually between 1.5 and 5) of the lengths of the curves. An alternative method for selecting potential sets of curves for grouping with the advantage of not requiring an arbitrary threshold is to determine the neighbourhood of each curve based on local constraints. This procedure is often done for points [1] or straight lines [6] by triangulating the data. The resulting triangle edges define the neighbourhood relationships. In our case, we triangulate the endpoints of the curves. In addition, the triangulation is constrained so that the triangle edges do not cross any curves.

The greedy triangulation algorithm was used. Although it is possible to design a greedy triangulation algorithm with complexity $O(n^2 \log n)$ the simpler version we have implemented is $O(n^3)$ [12]. The triangulation is determined by generating

all $\binom{n}{2}$ potential edges between pairwise combinations of curve endpoints. Any edges that intersect the curves are removed. The remainder are then considered in order of increasing length. An edge is retained if it does not intersect any of the previously retained edges, otherwise it is discarded. The final set of retained edges makes up the constrained triangulation. Testing for intersection between potential edges and curves is an expensive process since each edge has to be tested for intersection with all the short line segments formed by all pairs of adjacent points within each curve. However, the process is speeded up by first checking the MBR of the curve. Only if the edge intersects the MBR are the line segments that form the curve tested for intersection.

3 Grouping operations

3.1 Continuation

Several techniques have been described for detecting the continuation of straight lines, i.e. colinearity [7]. Since the shape of straight lines is more restricted than curved lines the grouping rules can be much simpler. In contrast, detecting the continuation of curved lines needs to take into account the amount of bending and the possibility of structures existing at multiple scales. To some degree this has been addressed by the following methods.

Meer *et al.* [9] approximate curves in a pyramid by straight lines, and their rules for grouping curves involve the length of the straight line separating their endpoints, the difference in orientation between the two line segments to be grouped, and the similarity of curvature of the two curves. The curvature is based on the change in orientation of the last three line segments at the end of each curve. Mohan & Nevatia [11] also use the Euclidean distance between the ends of the curves. Their other factor is the difference in angles between the connecting straight line and the tangents to the endpoints of the curves. Dolan & Riseman [3] represent all curves by conic splines. Colinearity between pairs of curves is based on the distance between their endpoints, the angular difference between the endpoint tangents, and the percentage overlap of the curves.

In a similar vein to the above techniques we evaluate curvilinearity based on the gap distance relative to the lengths of the curves, and the amount of bending of the completing curve. Rather than complete the curves with a straight line they are interpolated as described previously. It is the length of this interpolated section that is used rather than the Euclidean distance between the curve endpoints. The curvilinearity measure takes the form

$$\frac{l_{interp} \kappa_{max}}{l_1 + l_2}$$

where l_{interp} is the length of the interpolated curve bridging the gap, and l_1 and l_2 are the lengths of the two curves being grouped, and κ_{max} is the maximum curvature of the interpolating curve. Perfect curvilinearity produces a response of 0. Non-perfect groupings produces larger positive values. Advantages of this measure are that it allows long curves to bridge larger gaps, and that it is scale independent.

However, it has the disadvantage that all collinear curves (i.e. perfectly aligned straight lines) produce a zero response irrespective of their gap.

3.2 Parallelism

Determining parallelism for straight lines mainly involves their overlap and orientation [7]. Since the shapes of arbitrary curves are more variable the grouping rules need to be more complex. Bergevin & Levine [2] approximate curve sections by circular arcs. Parallelism can then be based on overlap, radii, and circle centres. However, arbitrary curves may not be readily decomposed into circular arcs.

More generally, Mohan & Nevatia [11] search for symmetric pairs of curves, which are a superset of parallel pairs. Curves are segmented at curvature extrema and all combinations of curve sections generate a symmetric axis. This is pruned by eliminating axes formed by pairs of curves with dissimilar lengths or little overlap. The most significant axes are selected based on a weighted sum of the following measures: the length of the contours covered by the axis; the ratio of the axis length to the two curves; the similarity of the length of the two curves; the amount of skew between the curves; parallelism between the curves; parallelism between the ends of the curves; and the length of the axis. An advantage of this approach is that sections of curves are grouped as wholes, thereby reducing computation compared with a point by point analysis.

We substantially adopt Mohan & Nevatia's approach. Curves are segmented at singular points as described previously. The symmetry axis between two curve sections is defined as the locus of midpoints of the lines joining points at equal length ratios along the curves. Figure 1a shows two curves C_1 and C_2 of length s_1 and s_2 . A point x on curve C_1 is mapped to a point y on C_2 if and only if

$$\frac{a}{s_1} = \frac{b}{s_2}$$

where a is the length of the section of curve C_1 up to x , and b is the length of C_2 up to y . Since we are detecting parallelism rather than skew the symmetry axis is not of direct interest. Rather it can be used to assess the potential parallelism by enabling two measurements to be made, namely the separation between the curves and the amount of overlap of the curves onto the symmetry axis.

Since the curves have been segmented into simple sections with monotonically increasing or decreasing curvature it is generally adequate to test the separation at just the endpoints and midpoint. If the endpoints of the curves are skew then the joining lines that define the symmetry axis will overestimate the separation of the curves. The true separation of the curves at a point on the axis is found more accurately by starting on the curve at the point associated with the axis. The curve is tracked along until a local minimum distance from the axis point is found. Usually this is also a global minimum and the new line is normal to the curve (figure 1b). For each endpoint this process is applied to both curves and the smaller of the two minimum distances is taken. For the midpoint only one curve need be examined and the smaller of the local minima found when tracking in either direction is kept. The ratio of the minimum and maximum values of the three measured separation distances are used to define the degree of parallelism of the curves.

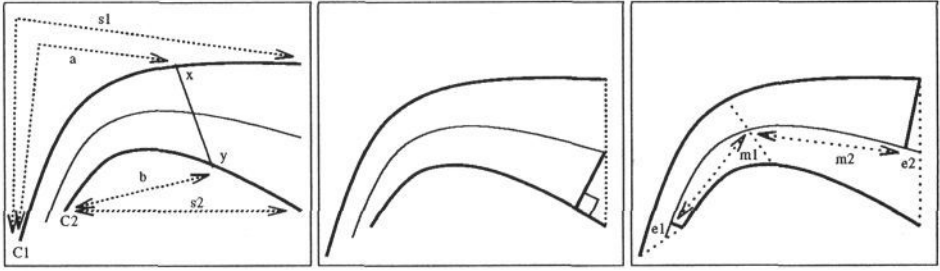


Figure 1: Calculating parallelism

Overlap of the curves is calculated by detecting the amount of overshoot of each curve when mapped onto the other. At each end of the symmetry axis the point closest to the endpoint of the curve which was not involved in the minimum separation described in the previous paragraph is determined. We call these points the secondary end points. Overlap is defined as the ratio of the length of the axis between the secondary end points to the length of the complete axis. This is approximated by the straight line segments from the axis endpoints to the secondary endpoints, and from the secondary endpoints to the midpoint of the axis as shown in figure 1c. Thus the overlap ratio is:

$$\frac{m_1 + m_2}{m_1 + m_2 + e_1 + e_2}$$

The parallelism of curves is sometimes split into two categories in which two identical curves are simply translated, or the curves alternatively shrink and expand with respect to each other so that sections of one are nested within the other. The first category are only approximately parallel. As the curves become closer the variation in the separation along the axis becomes more apparent. The techniques for measuring parallelism described above does not need to distinguish between the two categories.

The parallelism measure is calculated in a similar manner to curvilinearity, based on the gap distance relative to the lengths of the curves, and the degree of overlap between the curve:

$$\frac{l_{gap}}{l_1 + l_2} (1 - r_{overlap})$$

where l_{gap} is the distance between the curves, l_1 and l_2 are the lengths of the two curves being grouped, and $r_{overlap}$ is the overlap ratio calculated as above. A reasonable approximation for l_{gap} is the sum of the perpendiculars at either end of the curves. Perfect parallelism produces a response of 0, and non-perfect groupings produce larger positive values. As before, this measure allows long curves to bridge larger gaps, is scale independent, but produces a zero response for all perfectly aligned parallel curves irrespective of their gap.

In a similar manner to the curvilinear grouping, an alternative mechanism to distance thresholding for selecting curve sections for potential groupings is to triangulate the curve sections. A constrained triangulation can be formed as described

previously, where all curvature extrema and curve endpoints are taken as triangle vertices. Sections are only considered for parallel grouping if they are connected by a triangle edge.

3.3 Parallel Groupings

In addition to pairwise parallelism it may be useful to find larger groups of mutually parallel curves. This can be done by creating a graph whose nodes are the curves, and contains arcs between pairs of nodes if the two curves satisfy the parallelism relationship. The biggest set of mutually parallel curves is given by the largest maximal clique in the graph. A simple example of this is demonstrated in figure 2a which shows an Ehrenstein sun figure with some distorted and missing lines. We have set the constraint on allowable difference in orientation between parallel lines to be 45° . The largest maximal clique in the graph produces the lines shown in figure 2b. This procedure can be reapplied to find less significant parallel groups. Deleting the nodes comprising the largest maximal clique from the graph, the lines forming the next largest maximal clique are shown in figure 2c.

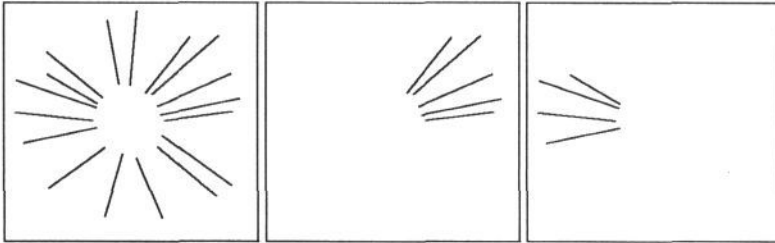


Figure 2: Ehrenstein sun figure and the two largest mutually parallel groupings

3.4 Proximity

Curves are considered proximal if they are near to each other. More specifically, we can divide proximal groupings into three classes of increasing specialisation: 1) any part of one curve is near to any part of a second curve – this includes a curved version of T junctions; 2) only their endpoints are near; and 3) the endpoints are close and the curves extend to a common point – this is also called cotermination. When the endpoints are close (classes 2 and 3) they can be grouped to form a virtual line (a well known example is the Ehrenstein sun figure).

We apply the following method for detecting virtual lines using triangulation. The problem is simplified by considering the triangulation of the curve endpoints since the virtual lines will be approximately outlined by the triangle edges. One way of hypothesising virtual lines with specific shapes from amongst the edges is to apply the Hough Transform. Edges are accumulated, and peaks in accumulator space define the parameters of likely virtual lines. Votes can be made inversely proportional to the lengths of edges. This ensures that close endpoints are considered more significant than distant ones. It is advantageous to apply the Hough Transform to the triangle edges rather than to the curve endpoints since

the edges contain more information. First, the edges provide orientation information, thereby simplifying and improving the detection of the virtual lines. Second, they also take into the account lines which cross the virtual line, thereby weakening the illusion of the virtual line. This effect cannot be easily included if the endpoints are grouped directly. In comparison, such lines prevent some triangle edges forming, and therefore the Hough Transform will produce a weaker peak in the accumulator.

Work is underway on detecting coterminations. Currently, proximity is not incorporated, and all curves are extrapolated at each end using the endpoint and the three adjacent singular points. Using a graphics routine the extrapolated curves are written into an image. Coterminations are identified as curve intersections, and are found by searching for peaks in the image. Since the confidence in the extrapolation decreases with distance from the original curve endpoint, the plotted values are made inversely proportional to the distance. To allow for inaccuracies in the intersections the image is blurred before searching for peaks.

4 Examples

Figure 3a shows a test set of curves which will be used to demonstrate the different grouping operations. The natural scales of these curves are shown in figures 3b-f, where the thickness of the curves is proportional to the amount of smoothing (for more details see Rosin [13]). For display purposes the curves have been separated into five sets (the maximum number of natural scales for any curve).

The constrained triangulation of the curves to determine their neighbourhoods is shown in figure 4. The method of extending the curves prior to their smoothing ensures that their endpoints are fixed [14]. Therefore, a curve at all its natural scales will have identical endpoints. Triangulation of the curves at different scales will not necessarily be identical since, in general, the curve shifts locally. This affects the constraint on the triangle edges not intersecting the curves. However, the triangulation will be similar, and any differences will be local. Therefore, for simplicity we only perform triangulation at one scale (the unsmoothed data) and apply the neighbourhood relationships at all scales.

An example of curvilinearity grouping from the end of one curve is shown in figure 5a. The curve of interest is drawn bold, and the end being grouped is circled. Grouping is restricted to the ends of those curves which are connected by a triangle edge to the endpoint of the curve of interest. The interpolated curves are shown in figures 5b-e in decreasing order of merit.

To detect parallel structures we segment the curves at curvature extrema which are then triangulated. This can be expensive if triangulations are performed for all combinations of curves at different natural scales. Here we restrict the triangulation to a single set of natural scales. The triangulation of the coarsest natural scales is shown in figure 6a. Maxima, minima, and zero-crossings of curvature are marked by boxes, circles, and crosses respectively. Figure 6b shows a selected curve section (drawn bold) and the potential parallel curve sections connected by at least one triangle edge (drawn in grey). The two curve sections producing the best parallel measure are shown in decreasing order in figures 6c&d. The thin lines are calculated by searching for the local minimum distances between the curves

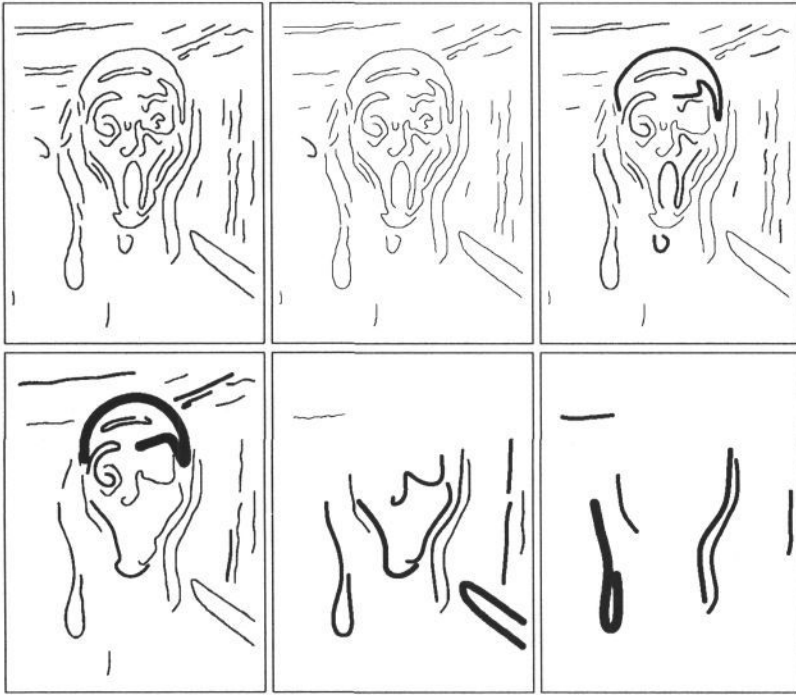


Figure 3: (a) original curve; (b)–(f) natural scales



Figure 4: Constrained triangulation of curves at original scale

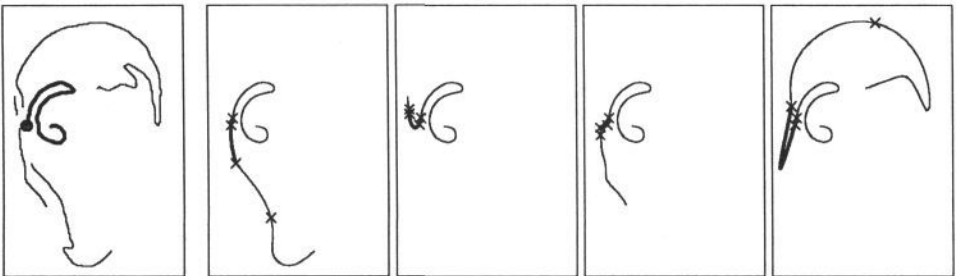


Figure 5: (a) one curve (bold) and its neighbours; (b)–(e) best continuations

and indicate the amount of overlap. The remaining potential curve sections had zero overlap.

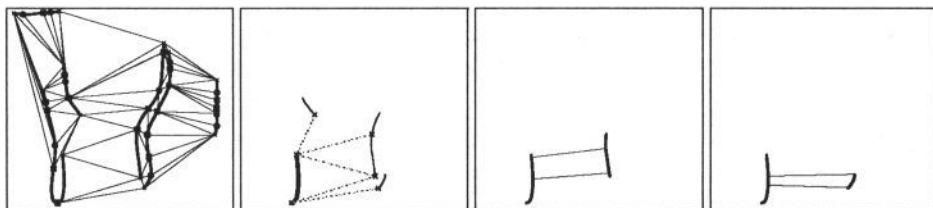


Figure 6: Detecting parallel groupings

To determine virtual lines the triangle edges in figure 4 are input to the Hough Transform (bin sizes are $\rho = 8$, $\theta = 9^\circ$). Figure 7 shows the virtual lines (drawn bold) corresponding to the two largest peaks. The endpoints of the triangle edges (and therefore also the curves) that gave rise to them are circled.



Figure 7: Virtual lines (bold) generated by triangle edges (circled)

Finally, the detection of coterminations is shown for the image in figure 8a. Results are only shown at a single natural scale; the curves at this scale (drawn bold) with their extrapolations are shown in figure 8b (the distance weighting is not shown). The extrapolation image is averaged with a 5×5 window and the 10 largest peaks (shown by circles with radii proportional to peak size) are overlaid with the curves in figure 8c.

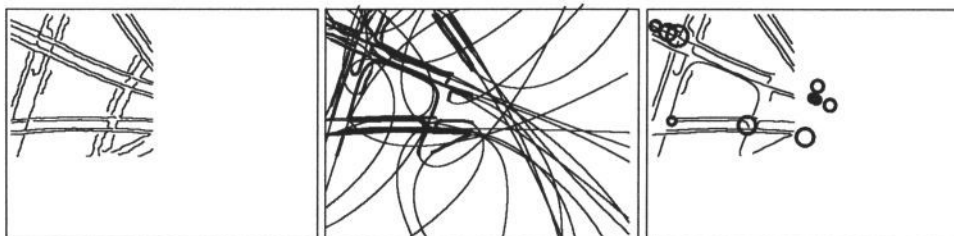


Figure 8: Detecting coterminations

5 Future work

We have described methods for performing various curve grouping operations. Future work will concentrate on analysing their performance in terms of robustness over a range of data, and their relationship with human visual processing [4].

References

- [1] N. Ahuja and M. Tuceryan. Extraction of early structure in dot patterns. *CVGIP*, 48:304–356, 1989.
- [2] R. Bergevin and M.D. Levine. Extraction of line drawing features for object recognition. *Pattern Recognition*, 25:319–334, 1992.
- [3] J. Dolan and E. Riseman. Computing curvilinear structure by token-based grouping. In *Proc. CVPR*, pages 264–270, 1992.
- [4] J. Elder and S. Zucker. A measure of closure. Technical Report 93-2, McGill Research Centre for Intelligent Machines, McGill University, March 1994.
- [5] H. Freeman. Shape description via the use of critical points. *Pattern Recognition*, 10:159–166, 1978.
- [6] D.P. Huttenlocher and P.C. Wayner. Finding convex edge groupings in an image. *Int. J. of Computer Vision*, 8:7–27, 1992.
- [7] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer, 1985.
- [8] D.H. Marimont. A representation for image curves. In *Proc. AAAI*, pages 237–242, 1984.
- [9] P. Meer, C.A. Sher, and A. Rosenfeld. The chain pyramid: Hierarchical contour processing. *IEEE Trans. PAMI*, 12:363–375, 1990.
- [10] J.E. Midgley. Isotropic four-point interpolation. *CGIP*, 11:192–196, 1979.
- [11] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *IEEE Trans. PAMI*, 14:616–635, 1992.
- [12] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.
- [13] P.L. Rosin. Representing curves at their natural scales. *Pattern Recognition*, 25:1315–1325, 1992.
- [14] P.L. Rosin. Non-parametric multi-scale curve smoothing. *Int. J. Pattern Recognition and AI*, 1994.
- [15] P.L. Rosin and G.A.W. West. Detection and verification of surfaces of revolution by perceptual grouping. *Pattern Recognition Letters*, 13:453–461, 1992.
- [16] E. Saund. Labelling of curvilinear structure across scales by token grouping. In *Proc. CVPR*, pages 257–263, 1992.