

REDUCING THE NUMBER OF POINTS IN A PLANE CURVE REPRESENTATION

Terry J. Deveau

Halifax Fisheries Research Laboratory
Department of Fisheries and Oceans
Scotia-Fundy Region
P.O. Box 550
Halifax, N.S., Canada, B3J 2S7

Telephone: (902) 426-4224

ABSTRACT

The most common representation of a plane curve in a digital computer system consists of an ordered set of points implicitly joined by straight line segments. An algorithm is presented which attempts to reduce the number of points in this representation while retaining the curve's salient features. The algorithm is compared with two other published procedures.

INTRODUCTION

General Problem

When a plane curve is represented by a subset of its points, two requirements arise immediately:

- (a) the number of points included in the subset should be minimized to lessen the resources required in storing and processing the curve.
- (b) the subset selected should be such that the polygon formed by joining the member points with straight line segments matches the original curve within some acceptable tolerance.

Note that in other contexts the 'straight line segments' of (b) could be replaced with 'circular arcs', 'cubic splines', 'conic segments', or some other curve family; this work deals only with the straight line segment case.

Clearly, (b) can be satisfied by increasing the size of the subset; we need some way to find the smallest sufficient subset. In developing such an algorithm we must be mindful of a third constraint:

- (c) the processing required to select the reduced points should be kept to a minimum.

Application to Cartography

Consider the problem of rendering the same coastline repeatedly on a number of separate maps as an example of the practical aspects of this problem. The coastline may have been digitized at high resolution and stored in a computer data base. The required maps could be at various scales, requiring that only a subset of the data base be used for each map, to provide the proper generalization of coastline features and effective use of the plotting hardware.

To avoid the cost of having several copies of the coastline at different resolutions stored in the data base, an efficient reduction algorithm is required which can produce the desired subset at only a small additional cost over that of a straight sequential read.

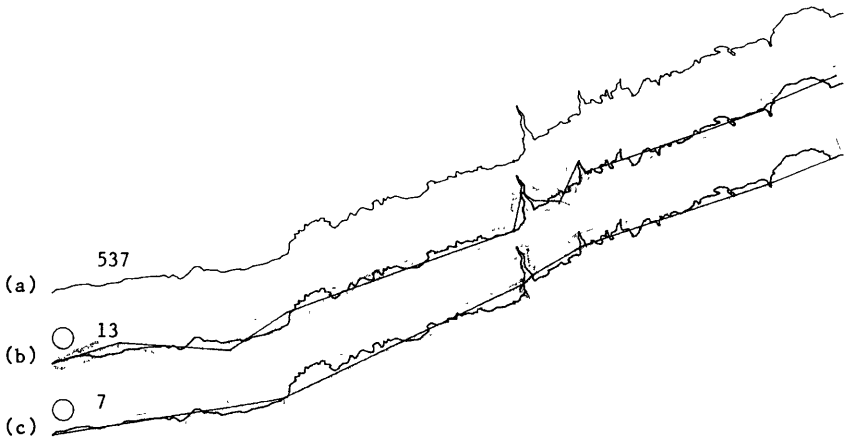


Figure 1. (a) A section of the 500 fathom depth contour off the Scotian Shelf. The number at the left is the number of points in the line. (b) The same contour overlaid with a reduced line fit using a centred band ($NW=1$), the tolerance used (TOL) is indicated by the diameter of the circle at the left and the width of the shaded band; (c) shows a second fit using a floating band ($NW=10$).

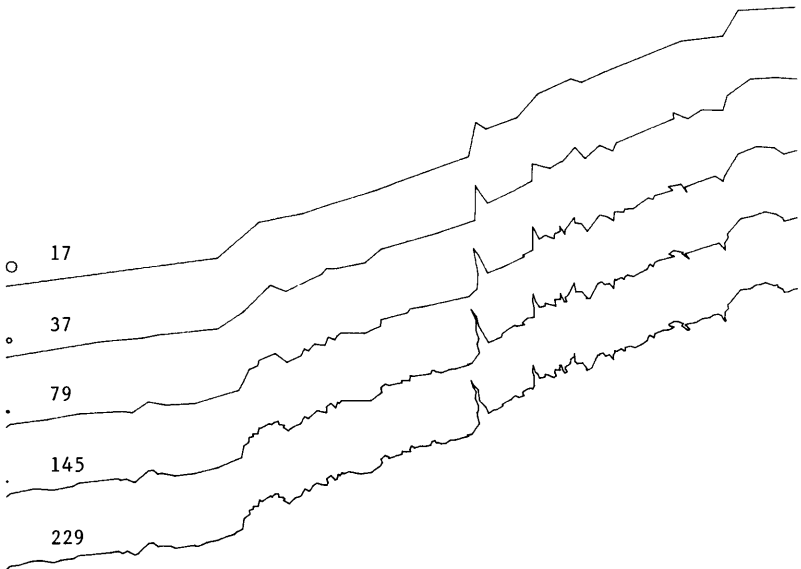


Figure 2. These lines have been fit to the same depth contour shown in figure 1. The same parameters were used as in line (c) of figure 1 except for TOL which was halved for each successive fit.

Additional Requirements

After consideration of the practical problems associated with automatic cartography we find that there are additional features we would like to see in the reduction algorithm:

- (d) works with open as well as closed curves;
- (e) avoids systematic deviations from original curve (even within tolerance);
- (f) prefers members of original set for inclusion in the reduced set (i.e. interpolation minimized);
- (g) promontories not blunted indiscriminately;
- (h) eliminates small closed curves in the furtherance of cartographic generalization (must use a test which produces results in line with the decisions a cartographer would make).

With the exception of (h), these further requirements need not compromise the algorithm's ability to be used for non-cartographic work (as long as attention to them does not make the algorithm significantly more expensive to use).

DEVELOPMENT

General Solution

The first step in developing the required algorithm is to quantify what is meant in (b) by 'match within some acceptable tolerance'. One way is to require that a band of width TOL, centred on the reduced polygon, include within it the original curve. The value TOL then becomes a parameter driving the reduction algorithm. Algorithms which use this fit criterion will be called 'centred band' algorithms.

For cartographic applications however, a second (looser) definition is often useful: for each segment of the reduced polygon, some band of uniform width TOL must exist which contains both the polygon segment and that section of the original curve. This will be referred to as the 'floating band' criterion; it allows very narrow (but long and straight) anomalies to be removed when reducing the curve (e.g. a coastline broken by a small river).

The floating band criterion also allows fewer polygon segments to represent the curve to the same (numerical) tolerance than is allowed by the centred band approach. An illustration of the two methods is shown in figure 1 and the effect of various values of TOL in figure 2.

In consideration of (c) we want to process the points in sequential order and compute the reduced set during a single pass. In general terms, the solution will be obtained by finding a band which contains two consecutive points in the original curve, then adjusting this band for each subsequent point so that it includes all points under consideration. When a point is found for which such an adjustment is not possible, the first and last points which did fit are used as vertices of the reduced polygon. The process is then repeated starting with the latest polygon vertex.

Practical Algorithm

The algorithm presented here uses the first point to fix one degree of freedom of a line through the centre of the tolerance band, leaving a single degree of freedom for fitting the band to subsequent points. That freedom can be viewed as the angular orientation of the band.

Since the band's orientation will be constrained within two extreme angular positions, the centre lines of these two limiting positions are

maintained by the algorithm. Both lines pass through the starting point and form the edges of an acceptability wedge. Each subsequent point must come within TOL/2 of this wedge, otherwise the last point to fit will become a new vertex in the reduced polygon. One or both edges of the wedge will usually be adjusted with each point until a new vertex is required.

This approach allows the new vertex to fall near one edge of the tolerance band or anywhere inside it. The initial vertex however, has been restricted to the centre of the band. This is halfway between the criteria defined above because the tolerance band is centred at one end and floating at the other.

If instead of using the last point to fit as the new vertex, we use the nearest point to it which is in or on the wedge, the new algorithm becomes a strictly centred band procedure.

A floating band algorithm can also be derived from the above by keeping track of several wedges simultaneously, in addition to the wedge starting at the last polygon vertex. These additional wedges can start around the circumference of a circle of diameter TOL centred on the last vertex. When a point is found which is further than TOL/2 from one of the wedges, that wedge is dropped from the active set. Only when the last wedge is dropped does the last point which did fit become the new vertex.

The number of wedges (NW) is a parameter characterizing the algorithm. The author has implemented an algorithm which uses a centred band criterion when $NW=1$ and a floating band criterion for $NW>1$; this allows the best method to be selected for each application. Figure 5 shows several reductions of the same curve with different values of NW.

The centred band method outlined above can be improved slightly by using several extra wedges on the first segment and using the starting point of the last surviving wedge as the first vertex instead of the initial point of the curve. Credit for this idea belongs to Ivan Tomek who embodied something similar in his 'Algorithm F' (Tomek 1974).

Delays in Assignment of Initial Points

Features of a curve which have dimensions less than TOL are removed during reduction. If the entire curve (whether open or closed) has no dimension greater than TOL it seems reasonable to eliminate it completely. This consideration is independent of (h) above; here it is simply a matter of removing those curves which are so small that they would otherwise be reduced to just two points separated by less than TOL (or two identical points in the case of closed curves).

A practical algorithm can, in consideration of the above, delay treating the initial point of the curve until a point is found which cannot fit in the same circle of diameter TOL as all the preceding points. When this happens, the centre of the circle which has fit all points thus far can be used as the initial point.

If the final point in the curve is reached before the initial circle is exceeded, the entire curve can be dropped. Otherwise if the final point is intended to close the curve it should be replaced with the initial vertex of the reduced polygon.

Closed curves which exceed TOL in one dimension only would be reduced to degenerate curves of zero area and only two distinct points; these

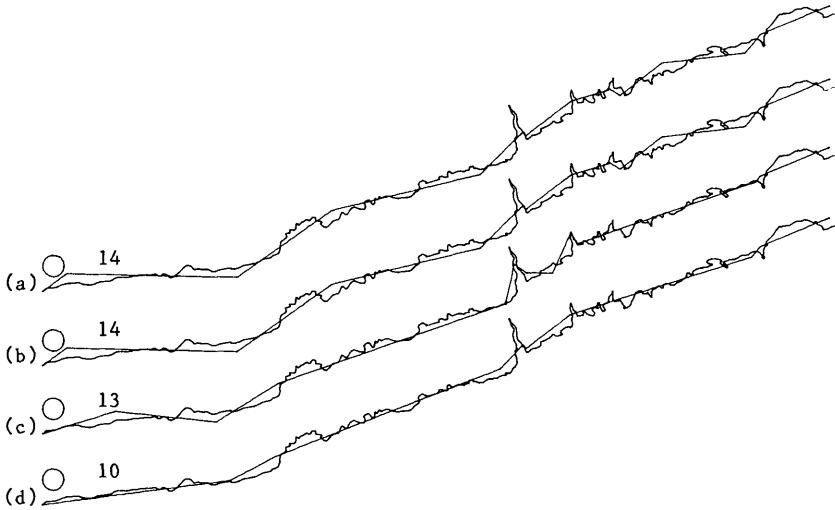


Figure 3. The 500 fathom depth contour is again fit with the same parameters as in figure 1 (b) except for bump angle (ANG). The effects of varying ANG can be seen here. Note that for a feature to be considered as a "bump" it must usually exceed TOL in both height and width. The values of ANG used here are (a) 0, (b) 60, (c) 120, and (d) 180.

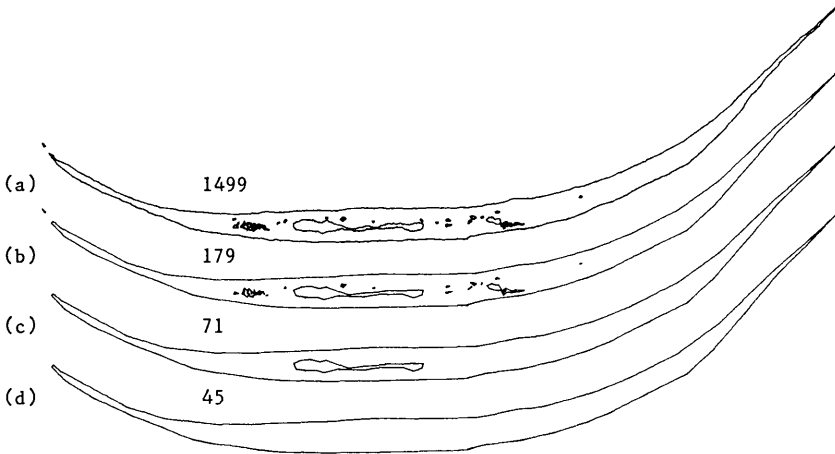


Figure 4. The coastline of Sable Island including some of its ponds. (a) Prior to reduction. (b) After reduction with FACTOR = 0.0; notice that the smallest features have been removed (no dimension > TOL). (c) After reduction with FACTOR = 1.0 and (d) with FACTOR = 2.0; use of FACTOR = 3.0 would completely remove the island itself. Other reduction parameters were held fixed in (b) through (d). As elsewhere, the number above each curve gives the number of points in the curve.

can also be removed (the next section will discuss other features which exceed TOL in only one dimension). For this reason (and to effect the improvement mentioned in the last paragraph of the preceding section) assignment of the initial polygon vertex can be delayed beyond the acceptance of the initial curve point to when the second polygon vertex is obtained.

Meeting Cartographic Requirements

The procedure outlined thus far will largely satisfy (a) through (f). Requirement (g) demands that we define another parameter to quantify what is meant by 'indiscriminately blunting promontories'. Lets say that if the vertex angle of a bump on the curve is less than ANG and its height and width are greater than TOL then the bump is a significant feature which should not be smoothed out in the process of finding the minimal polygon.

Figure 3 shows several reductions of a curve with different values of the parameter ANG. A value of 110° seems to produce good results for cartographic applications. Theo Pavlidis has also found a bimodality in vertex angles of text characters with a trough between 110° and 130° (Pavlidis 1983); this suggests that vertex angles on either side of this range are subjectively perceived as different features.

Because it is applicable only to cartography (h) can be looked at separately. Under this requirement we want to drop those closed curves which would appear too small on a map (in one or both dimensions). A simple test of the maximum and minimum extents of the curve would not do the trick however; a thin but curved feature (e.g. a horseshoe lake) would pass the test even if it was so thin on the map that both banks overlapped.

One test that was found to do the job fairly well is:

$$\frac{\text{area of closed curve}}{\text{perimeter}} > \text{TOL} \times \text{FACTOR} \quad (1)$$

where FACTOR is set to about one or two. Curves which fail the test are dropped. Figure 4 illustrates the control over auto-generalization provided by the adjustment of FACTOR in (1).

COMPARISONS

Implimentation

The author has implemented the algorithm outlined above as a FORTRAN subroutine suitable for use with a variety of application programs. A listing of this subroutine (which is too long to be included here) is available from the author.

This subroutine has been used for removal of redundant points from line-followed digitizer output, reduction of map data from a high-resolution data base to an appropriate level of generalization for larger scale maps, and matching general graphics output to the hardware resolution of a particular plotter.

The four parameters TOL, ANG, NW, and FACTOR provide the facility for tuning the method to the particular application without requiring maintenance of separate subprograms.

Douglas and Peucker Algorithm

This is basically a centred band algorithm (Douglas and Peucker 1973)

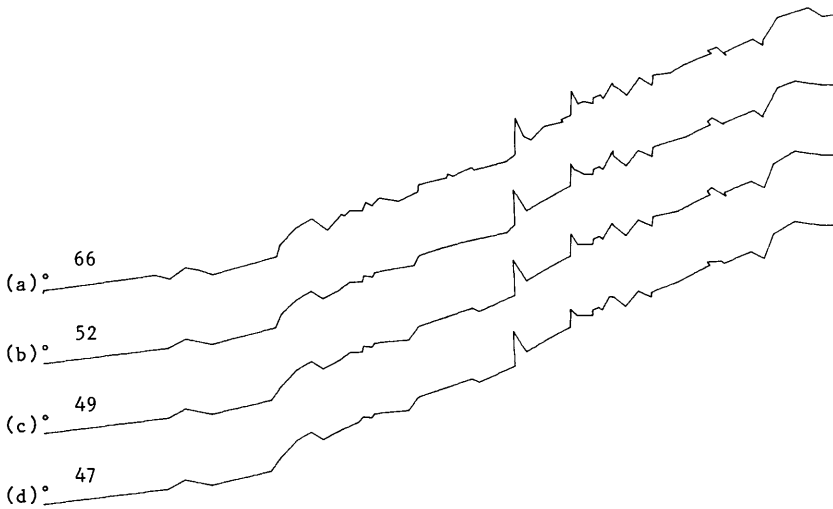


Figure 5. The effects of varying the number of wedges (NW) can be seen in these reductions of the depth contour shown in figure 1 (a). The first reduction (a) was done with NW=1 and uses a centred tolerance band; (b) NW=3, (c) NW=5, and (d) NW=10 use a floating band.

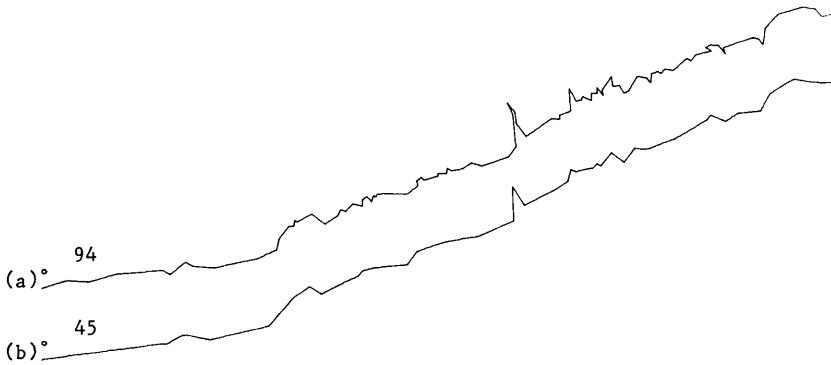


Figure 6. These reductions of the 500 fathom contour were done using the same tolerance as in figure 5; (a) was done using the Douglas and Peucker algorithm which uses a centred band. The Dettori and Falcidieno algorithm was used to produce (b), it uses a floating band.

with their 'offset tolerance' corresponding to half the band width. It has received wide recognition among cartographers as it was the first published procedure which attempted to address their specific needs.

Figure 6 (a) shows a reduced line produced by this procedure, it can be compared with figure 5 (a) which is also a centred band reduction. Note that although the same tolerance was used in both reductions, the method described here was able to find a smaller polygon.

Dettori and Falcidieno Algorithm

Dettori and Falcidieno (1982) describe a floating band algorithm which employs the convex hull of a subset to decide if a new polygon vertex is needed. Unfortunately, the example FORTRAN subroutine included in their paper contains several critical logic errors. Substantial debugging was necessary before the example subprogram accurately implemented the algorithm they presented.

This done, the method worked very well; as can be seen in figure 6 (b). This reduction was done at the same tolerance as that of figure 5 (d) and their method found the smaller polygon. Of course our use of the bump angle test (controlled by ANG) means that extra (cartographically significant) points will be included beyond those required simply to satisfy the tolerance criterion.

Objective Comparison

To provide an objective basis for comparison, a coastline file of 13,216 points was processed by these algorithms with equivalent settings of their control parameters. The input data included 794 coastal islands as well as a single 7437 point coastline curve.

The auto-generalization feature was not used for this comparison (FACTOR = 0.0) but 750 islands were eliminated by our algorithm because they were smaller than TOL in one or both dimensions.

The results are listed in table I. A CPU second is a measure of the computing effort used to perform the fit. The last two columns show the results of a second comparison run on the coastline segment only (no islands).

TABLE I - Comparison Data

<u>Method</u>	<u>Points Output</u>	<u>CPU sec.</u>	<u>No Islands</u>	<u>CPU sec.</u>
NW = 1	667	57.859	566	25.119
NW = 10	481	53.692	394	31.032
Douglas and Peucker	2766	58.936	909	25.339
Dettori and Falcidieno	2015	61.198	369	26.561

A more thorough and detailed comparison of these and other reduction procedures is in preparation and will be published separately.

CONCLUSIONS

The algorithm presented here provides extended control over the reduction process via four control parameters. It has several features of particular relevance to automatic cartography and provides a useful degree of auto-generalization. The flexibility afforded by this procedure allows it to serve diverse applications where a single algorithm may have been considered inadequate.

The techniques used here can also be used with non-linear piecewise

curve fitting. Circular arc fitting could be especially useful as some plotting hardware now accepts circular arc input.

The problem of auto-generalization also needs much more work with such things as merging nearby islands, deleting branching rivulets, and either widening or eliminating narrow channels offering a particularly tough challenge.

ACKNOWLEDGEMENTS

The author is indebted to Dr. Robert Mohn for his valuable suggestions and comments.

REFERENCES

Dettori, G. and B. Falcidieno 1982, An Algorithm for Selecting Main Points on a Line, Computers and Geosciences, Vol. 8, No. 1, pp 3-10

Douglas, D. and T. Peucker 1973, Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature: The Canadian Cartographer, Vol. 10, No. 2, pp. 112-122

Pavlidis, T. 1983, Curve Fitting with Conic Splines, ACM Transactions on Graphics, Vol. 2, No. 1, pp 1-31

Tomek, I. 1974, Two Algorithms for Piecewise-Linear Continuous Approximation of Functions of One Variable, IEEE Transactions on Computers, Vol. C-23, pp 445-448