Cartographic Algorithms: Problems of Implementation and Evaluation and the Impact of Digitising Errors

M. Visvalingam and J.D. Whyatt*

Abstract

Cartographic generalisation remains one of the outstanding challenges in digital cartography and Geographical Information Systems (GIS). It is generally assumed that computerisation will lead to the removal of spurious variability introduced by the subjective decisions of individual cartographers. This paper demonstrates through an in-depth study of a line simplification algorithm that computerisation introduces its own sources of variability. The algorithm, referred to as the Douglas-Peucker algorithm in cartographic literature, has been widely used in image processing, pattern recognition and GIS for some 20 years. An analysis of this algorithm and study of some implementations in wide use identify the presence of variability resulting from the subjective decisions of software implementors. Spurious variability in software complicates the processes of evaluation and comparison of alternative algorithms for cartographic tasks. No doubt, variability in implementation could be removed by rigorous study and specification of algorithms. Such future work must address the presence of digitising error in cartographic data. Our analysis suggests that it would be difficult to adapt the Douglas-Peucker algorithm to cope with digitising error without altering the method.

1. Introduction

One of the main benefits of automation in cartography is the scope that it offers for the removal of spurious variability introduced by the subjective decisions of individual cartographers. Many of the benefits accredited to quantification are also attributed to computerisation. It is assumed that a tested program will produce objective, consistent and predictable results. However, it is a fallacy to assume that it would continue to produce the same results in a different computing environment. No doubt the reliability of a piece of software may be tested using benchmarks. However, this assumes that the benchmark has been rigorously formulated. This is no mean task. Forrest1 examined some of the complexities involved in the implementation of geometric algorithms, using detection and computation of line intersections as examples. Forrest examined how inadequate consideration of special geometric cases and of the precision, method and order of computation can yield incorrect or inconsistent results when primitives for line detection and intersection are used within point-in-polygon tests using the parity algorithm. In comparison, the specification of the Douglas-Peucker algorithm[2] is somewhat more complex and the incomplete description of the original algorithm provides ample scope for alternative interpretations and implementations. Also, the algorithm can produce variable results even when subjected to precise calculation because of the nature of digital cartographic data.

The aim of this paper is to explore the potential scope for variability in the interpretation, implementation and evaluation of cartographic algorithms, using the Douglas-Peucker algorithm as an example. Unless the scope for variability is recognised, consciously identified through systematic testing procedures and standardised, it would be difficult for researchers in digital cartography to accept and utilize each other's conclusions about cartographic generalization with much confidence. This paper also identifies another major source of concern, namely the inadequate consideration of digitising errors in spatial data processing.

2. Background

The Douglas-Peucker algorithm enjoys special mention within cartographic literature and has been widely adopted within mapping software and GIS. It has been promoted as "mathematically and perceptually superior" to other line simplification algorithms by McMaster[3]. Although others have provided anecdotal evidence to the contrary (see review in Visvalingam and Whyatt[4]), leading researchers in cartography and GIS single out this algorithm for special mention. For example, Goodchild[5] regarded it as one of the standard methods for spatial data analysis. The status of this algorithm has encouraged others such as Buttenfield[6] and Jones and Abraham[7] to apply it outside the narrow problem of line simplification without prior independent evaluation.

In the current still relatively low state-of-the-art of digital cartography it is necessary to retain a more critical frame of mind and pursue independent evaluations prior to adoption of algorithms and their implementations. Previous evaluations of the Douglas-Peucker algorithm, including those by McMaster, have tended to rely on perceptual and mathematical comparisons of the output line with the original input, i.e. on the use of black-box methods. Perceptual studies have relied on visual comparison of the original and

*Cartographic Information Systems Research Group
Department of Computer Science
University of Hull
HULL HU6 7RX, UK

North-Holland
Computer Graphics Forum IO (1991) 225-235

filtered lines whilst mathematical comparisons have been based on gross measures, such as of vector and areal displacement, which have been questioned elsewhere[8]. Visvalingam and Whyatt[4] used visualization techniques for the evaluation of the algorithm. Instead of relying on a passive visual assessment of simplified lines, i.e. the output, they used alternative visualizations of tag values associated with vertices and visual logic to pursue hypotheses and draw conclusions about the algorithm, its underlying assumptions and their implications. They made some critical observations about the algorithm. This paper examines some of the problems facing the implementation of this algorithm as a computer program.

3. Scope for Variability in Implementation

One of the reasons for the popularity of the so-called Douglas-Peucker algorithm is its elegant formulation. The numerous published accounts of this algorithm have not exposed, let alone discussed, many awkward decisions involved in the expression of this algorithm as a computer program. Consequently, there exist different interpretations and implementations of the algorithm, producing different results. Further, not all implementors and users of cartographic software appear to be aware of the accuracy problems involved in computation. Equally, no attention has been paid to the existence of digitising errors when formulating algorithms. It appears that such errors can only be dealt with in an ad-hoc manner when using this simple and elegant but perceptually inadequate procedure.

3.1. Variability in Interpretation

Since the original description of the Douglas-Peucker algorithm was unclear, others have offered their own descriptions; some of which appear to be erroneous. Our interpretation of this method is as follows: A base line, known as an anchor-floater line, is used to connect the first and last points of a line. Perpendicular offset values from this line are calculated for all intervening points. If the furthest point from the base line falls within some pre-defined tolerance, then it is assumed that the original line may be approximated by a straight line segment. If the offset of the furthest point exceeds the tolerance, then the original line is subdivided at this point, and the two parts of the original line are treated as independent lines which are subjected to the same process for simplification or subdivision. Our interpretation corresponds to the method of iterative end-point fit described by Duda and Hart[9], who stated (on p. 373) that the method was first suggested by G. E. Forsen. The most detailed description of the algorithm was provided by Ramer10, who described it as an iterative procedure for approximating plane curves by a small number of vertices lying on the curve. His illustrations included a scale-related simplification of the coastline of Seward Peninsula.

3.2. Variations in Implementations

Different implementations of the Douglas-Peucker algorithm produce different results since programmers have coped with exceptional geometric cases and numeric problems in different ways. Some of these problems are described below and are illustrated using output from the programs of Douglas[11], White[12] (comments indicate that the program was written by McMaster) and Wade[13]. We also include observations on results produced by GIMMS[14] and examine the implications of Ramer's analysis of special cases.

3.2.1. Special Geometric Conditions

a) Increasing Offset Values

Although offset values from the current anchor-floater line tend to decrease with progressive subdivision of lines, Peucker[15] noted that it was possible for offset values to increase with segmentation of a line. For example in Figure la, the first offset C-C' is smaller than subsequent offsets D-D' and E-E'. Both Douglas and Peucker[2] and Peucker1[5] envisaged that a pre-defined tolerance value would terminate the selection and thus the further subdivision of a line. Consequently, in Figure la, we would either retain or omit all of D, C and E. This provides a consistent, if not a desirable rule; for example, spikes are retained as a result. The latter could be removed through the decision to retain only those points whose offsets exceeded a given tolerance. For example, specifying a tolerance of 28 metres would result in the retention of points D and E only in Figure la. However, this rule would pose equally difficult problems in other circumstances; specifying a tolerance of 28 metres would result in the retention of point D without point C in Figure 1b. The resulting simplification is inappropriate.

The rule, implied by Douglas and Peucker, would be honoured if the algorithm was repeatedly applied each time a line had to be filtered; the programs by Douglas and White are used in this way. However, this is very wasteful of computing resources and it is more efficient to apply the algorithm just once to assign tag values (see below) to points. Subsequent filtering of lines would then rely on comparing these pre-computed tag values against a given threshold or tolerance. This idea was first used in GIMMS14 in the GENERAL command, which is used to specify up to nine tolerance values, corresponding to decreasing levels of generalisation. These values are used to tag codes, in the range 1 to 9, to each vertex on the line. The start and end points of the input line are assigned the code of 0. When GIMMS subdivides a line at its maximum offset, it compares this offset against the given set of tolerance values, starting with the largest. If the offset exceeds this first tolerance value, then a code of 1 (first tolerance in list) is stored with the point. If the offset is less than the tolerance, it is tested against the second slightly smaller

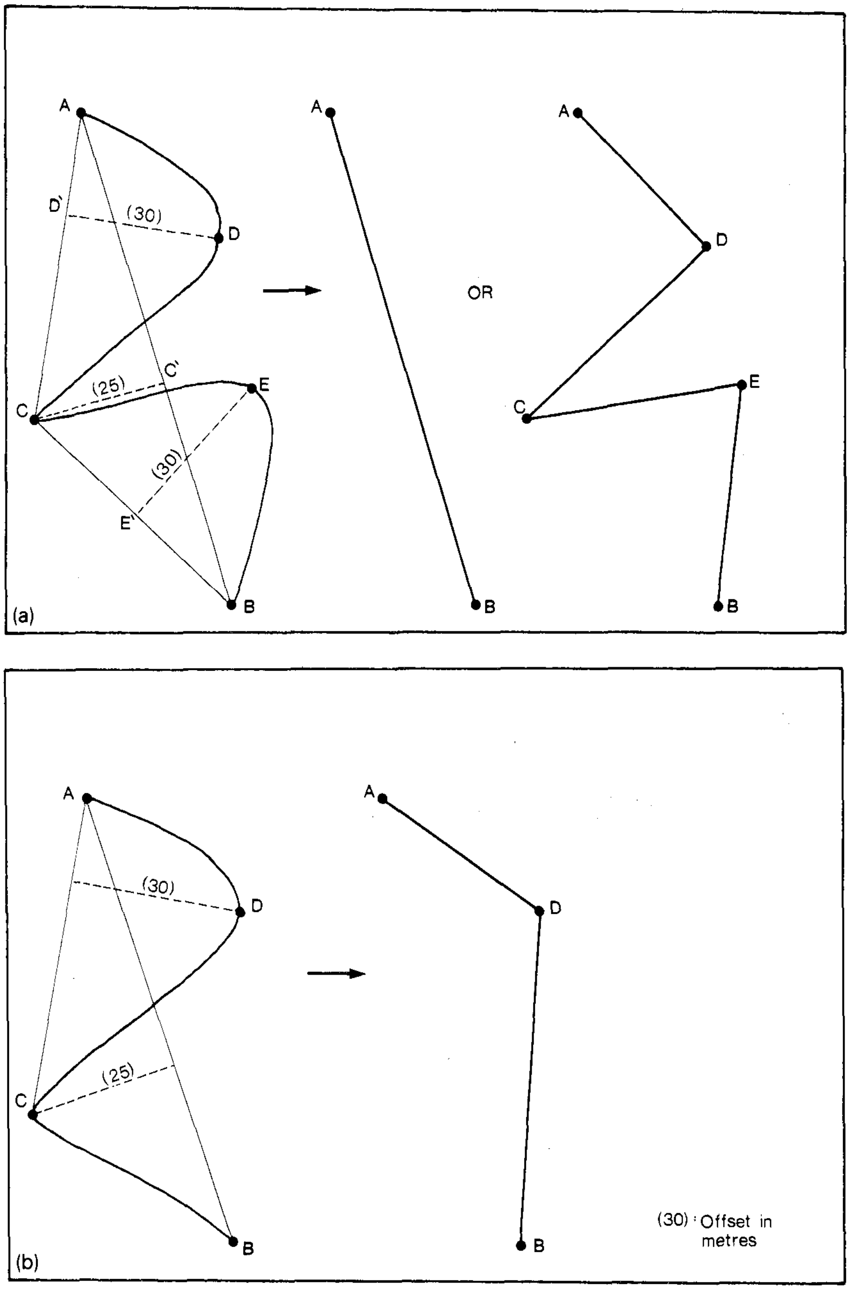

Figure 1. The problem of offset values increasing on segmentation of a line,

tolerance. The process repeats until the offset exceeds a tolerance value in the list; at which stage, the vertex is tagged with a number corresponding to the position in the list of this tolerance value. Note that by using this procedure it is possible to retain D and E as in Figure la, without retaining C. We are not suggesting that this is intrinsically wrong; we merely wish to point out that here is a case where different implementations can produce different results.

Wade[13] designed his implementation such that a line may be filtered at any scale at run time using any tolerance value. This requires that each vertex has associated with it a tag value which will normally correspond to the maximum perpendicular offset value which resulted in its selection. However, there is a need to ensure that the results produced are consistent with those produced by the original algorithm*. Wade's implementation therefore compares the offset value calculated for a given point with those for its

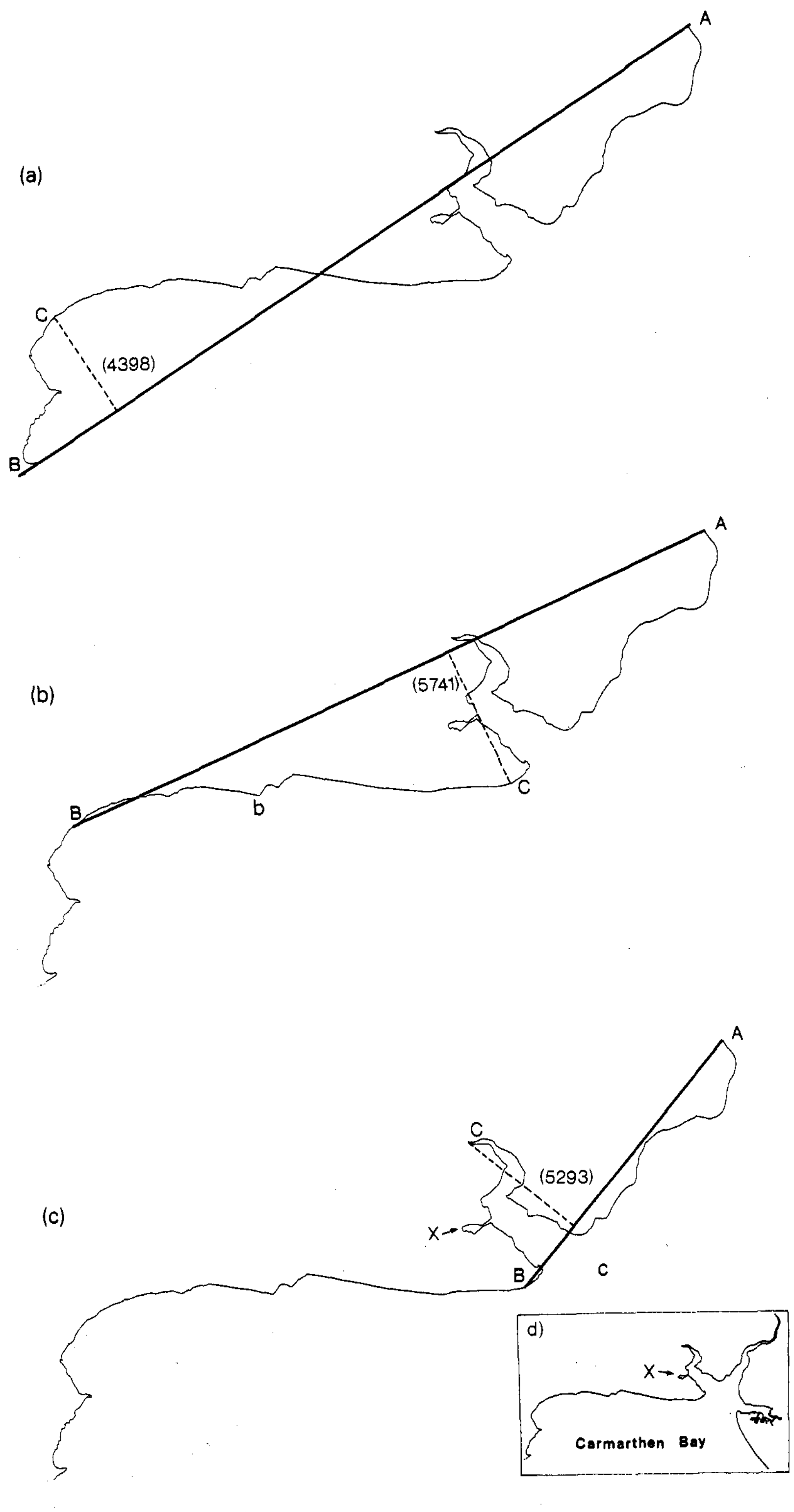

Figure 2. Illustration of increasing offset values along a section of coastline.

anchor and floater and records the smallest value as the tag value. Thus, in Figure 1a, points C, D and E would all have tag values of 25 metres. Whilst the possibility of this geometric case was noted by Peucker[15], it has been ignored perhaps because of the assumption that it is somewhat infrequent and exceptional. Figure 2, based on a section of the coastline of Carmarthen Bay in Wales, contradicts this assumption. This geometric case occurs fairly frequently along complex coastlines. For example, some 10% of the points on the coastline of Carmarthen Bay (Figure 2d) had their tag values adjusted. On randomly selected coastal sections of Cornwall, Cumbria and Sussex, 15-20% of points had to be adjusted.

Buttenfield[6] attempted unsuccessfully to use a number of statistics based on the algorithm for identifying line types; i.e. for pattern recognition. Although she used test lines which would have exhibited this geometric condition and included offset values in her set of statistics, she did not consider this problem in her analysis.

b) Overhangs

Figure 3 shows another geometric case which is not dealt with in the literature. Here we have a situation where a part of the line overhangs the anchor-floater line A-B. If we stuck rigidly to the wording of the algorithm, we should select point C. The programs by Douglas, White and GIMMS would select D, namely the point furthest from the infinite line of which the anchor-floater forms a part. Wade's program would choose E, the point furthest from the finite line A-B and more specifically B in this case. The choice of this critical point can influence the selection of some subsequent points; yet the implementation details remain arbitrary and variable.

c) Closed Loops

Different implementations use different ad-hoc rules when dealing with closed loops. Only Ramer10 and Douglas and Peucker[2] consider this special case. Ramer proposed that any two distinct vertices could be selected arbitrarily for the initial anchor and floater. He believed that the best choice would be two oppositely located extremal points since he believed that the algorithm would select these eventually anyway. In his algorithm he specified the choice of the highest left-most point and the lowest right-most point for these extremal points. Douglas and Peucker (p. 117) specified that where there are closed loops, the maximum perpendicular distance should be replaced with the maximum distance from that point. Wade's program takes this furthest point. White's program does not consider this case. The calculations, which assume an open line, would select the point furthest from the origin.

Both Ramer and White used consistent but arbitrary rules for splitting a closed loop. Douglas and Wade used a rule related to the configuration of points to subdivide the loop but retained the original anchor-floater, which need not be a perceptually critical point. If this furthest point was used as the new anchor-floater in place of the digitised point, and if the furthest point from this was then used to subdivide the loop (see Figure 4), then the implementation would become less arbitrary and would conform more to the spirit of the algorithm.

In Wade's program, the overhang and closed loop are treated as generically similar problems and are dealt with by one rule. The loop is a line which overhangs a point, a degenerate anchor-floater line. The selected point is tagged with the distance from this point. When the line overhangs the anchor-floater line, the maximum offset from the finite line is calculated where appropriate and the distance from either the anchor or the floater is used as the offset in the case of points which overhang this base line. The point with the largest offset is selected. Neither Ramer nor White considered overhangs and their methods are arbitrary.

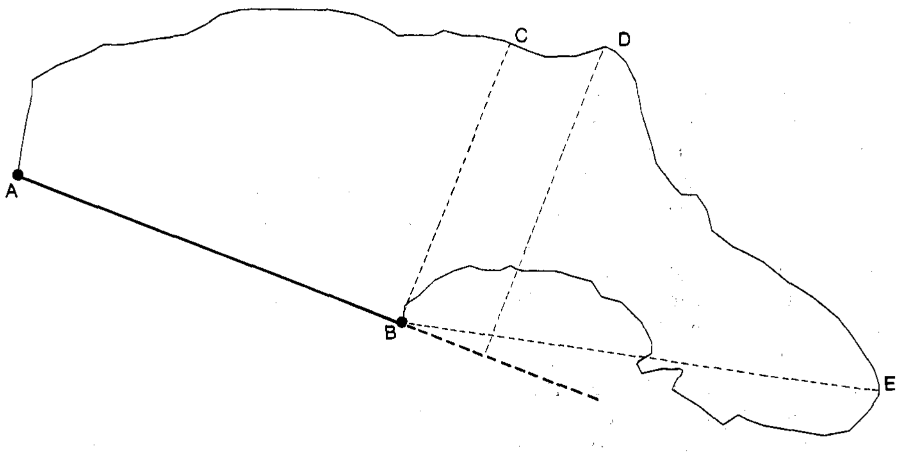

Figure 3. Lines overhanging the anchor-floater line (A-B).

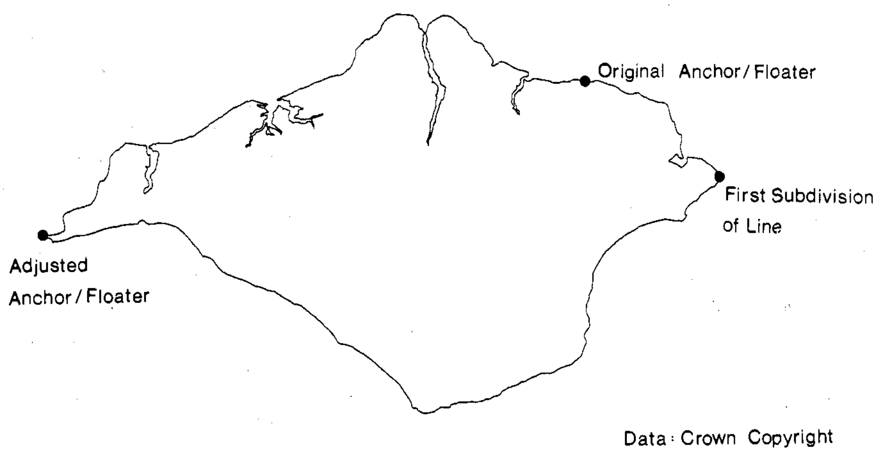

Figure 4. The need to adjust the position of the initial anchor-floater in a closed loop.

Douglas and Peucker have treated overhangs and closed loops as different problems, and have used different methods to cope with each case.

3.2.2. Numerical Problems

a) Accuracy of Computation

The FORTRAN programs by Douglas, White, and Wade use single precision REALS when computing offsets. Whilst double precision accuracy may be attained through the use of compiler options, we are unsure whether previous research has been based on programs compiled in this manner. Wade's program was so compiled for use in our previous evaluations[4]. Forrest[1] stated that Ramshaw (1982) had to adopt carefully tuned double and single precision floating point arithmetic to compute the intersection of line segments whose end points were defined as integers. Forrest' exclaimed "This is an object lesson to us all: constructing geometric objects defined on a grid of points, requiring ten bits for representation, can lead to double precision floating point arithmetic!".

Most evaluative studies do not cite the co-ordinates in use. We do not know whether the published test lines were in original digitiser co-ordinates or whether they had been converted to geographic references. British National Grid co-ordinates for the administrative boundaries of England, Scotland and Wales (digitised by the Department of Environment (DoE) and Scottish Development Department (SDD)) are input to one metre accuracy and require seven decimal digits for representation if we include the northern islands of Scotland. At the South West Universities Regional Computer Centre these co-ordinates have been rounded to 10 metre resolution; even this requires six decimal digits. Seamless cartographic files at continental and global scales use much larger ranges of geographic co-ordinates.

Most published simplification programs are written in FORTRAN and use single precision REALS for offset distances. Users of these programs should use compiler

Machine	Points	Calculated squares of offset values Single Precision	Double Precision
ICL 3980			
	(C)	28199.351562500000	28143.490838958319
	(D)	28171.789062500000	28143.490838961321
VAX 8200			
	(C)	28253.095703125000	28143.490838958267
	(D)	28165.806640625000	28143.490838958267
SEQUENT SYMMETRY			
	(C)	28145.100000000000	28143.490838961320
	(D)	28145.100000000000	28143.490838961320
SUN 3/60			
	(C)	28253.095703125000	28143.490838961323
	(D)	28165.806640625000	28143.490838961323

Table 1: The Precision of Calculations

NOTES

Offsets of points C and D from the anchor-floater line A-B as calculated using Wade's program. Points A, B, C and D are shown in Figure 5. The British National Grid co-ordinates (in metres) of the points are as follows:

Point A	238040	(x1)	205470	(y1)	ANCHOR
Point B	237890	(x2)	205040	(y2)	FLOATER
Point C	237810	(x3)	205320	(y3)	
Point D	238120	(x3)	205190	(y3)	

Note that the above co-ordinates may be used in conjunction with the expression presented in section 3.2.2a to check the tabulated results.

options for double precision arithmetic. The impact of using single precision arithmetic is demonstrated in Table 1. Even when compiled with the double precision option, the program by Douglas produces results which deviate significantly from those produced by others. The formula used to calculate the squares of offset values presented in Table 1 is as follows:

$$\lambda = \frac{(x1 * (x1 - x2 - x3) + x2 * x3 + y1 * (y1 - y2 - y3) + y2 * y3)}{(x2 - x1)^2 + (y2 - y1)^2}$$

$$x = x1 + \lambda * (x2 - x1)$$

$$y = y1 + \lambda * (y2 - y1)$$

$$dis = (x3 - x)^2 + (y3 - y)^2$$

In a recent debate on the accuracy of floating point calculations, Huggins[16] stated that the arbitrary-precision arithmetic language ‘bc’ could be used to obtain precise results. We used this UNIX utility to calculate offset values for points C and D. On the VAX 8200, SEQUENT SYMMETRY and SUN 3/60, bc returned identical values for these points:

C: 28143.490838958534 D: 28143.490838958534

Forrest[1] (p, 721) pointed out the well known fact that floating point calculations are still very much machine dependent. Machine dependency exposed further problems, which could be treated as problems of implementation but which are arguably more conceptual in nature as explained in the following sections.

b) **Equidistant Points from the Anchor-Floater Line**

The algorithm is based on the assumption that lines may be subdivided in an unambiguous manner using the maximum perpendicular offset. To our knowledge, the problem of two or more points being equidistant from the anchor-floater line has never been considered. Indeed, we only became conscious of this possibility when the same program yielded different results on ICL 3980 and SUN 3/60 computers. A sample problem is illustrated in Figure 5. Points C and D are equidistant from the anchor-floater line A-B. The inexact representation of floating point numbers results in C being selected on SUN workstations and D being selected on the ICL computer by the same program. With double precision arithmetic, the errors are negligible but are nevertheless sufficient to generate different results since published programs tend to use either a “greater than” or “less than” condition. GIMMS and the programs by Douglas and Wade select the first point from a set of identical offsets. White's program selects the last. The results therefore are variable and become dependent on the direction of digitising of lines. If, on the other hand, we select a point from this set at random, the procedure would become blatantly arbitrary. This problem poses other implications, which we will now examine in greater detail.

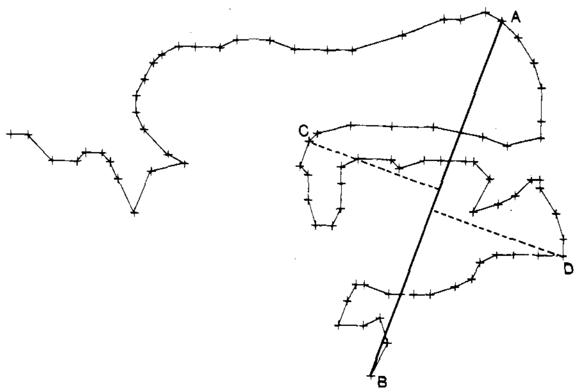

Figure 5. The problem of points (C&D) which are equidistant from the anchor floater-line (A-B).

3.3. Digitising Errors

Like most cartographic algorithms, the Douglas-Peucker algorithm does not fully address the issue of digitising errors. When estimating truth values, it is usually assumed that the true line (in this case the analogue line) lies within the error band of the digitised line (see Blakemore[17]). This band is also known as the Perkal epsilon band[18]. In his review on issues relating to the accuracy of spatial databases, Goodchild[5] indicated that researchers have proposed uniform, normal and even bimodal distributions of error across this band. This concept provides some basis for estimating the position of the true line at locations between digitised points. Here, we are merely concerned with the accuracy of digitised points. Whilst it is probable that operators digitise points along high curvatures more carefully than at intermediate positions, there is at present no sound basis for modelling the distribution of error along the line. As in the Circular Map Accuracy Standard, it is usual to assume a bivariate normal distribution of error when estimating the position of the true point. In the context of line simplification, absolute positional accuracy is less important than the relative position of points describing the shape of features along the line.

The DoE/SDD boundary data contain some gross digitising errors. For example, inlet X in Figure 2c does not feature on conventional Ordnance Survey 1:50 000 maps of the area. The data are also not very accurate where coastlines are convoluted. Even if we ignore these and other gross errors, such as spikes, there will always be an element of random error in digitised data. It is reasonable to assume that points digitised from 1:50 000 source material may only be accurate to within +/- 5 metres. This algorithm does not lead to a substantial accumulation of rounding errors, hence the numerical errors discussed earlier tend to be very small compared with digitising errors.

For the purposes of our argument, it is unnecessary to undertake an exhaustive evaluation of the consequences of digitising errors on the output of the Douglas-Peucker algorithm. We only need to explore some consequences in order to further our discussion. The rule used for the iterative subdivision of lines is the maximum distance from the anchor-floater line. Digitising errors affect its reliability in two ways. Firstly, it can alter the orientation of the anchor-floater line since the end-points are subject to error. Secondly, these errors have some impact on the use of the maximum distance as an indicator of perceptually critical points. We consider both these issues in turn.

Let us firstly reconsider the case of equidistant points. Some effects of digitising errors can be demonstrated using Figure 6a, in which points C, D and E are equidistant from line A-B. Digitising error implies that the orientation of the true line would deviate from the line A-B. Offset values from the true line would no longer be equal as shown in Figures 6b and 6c. Seen in this context, the selection of the first or the last equidistant point must be recognised as an arbitrary decision.

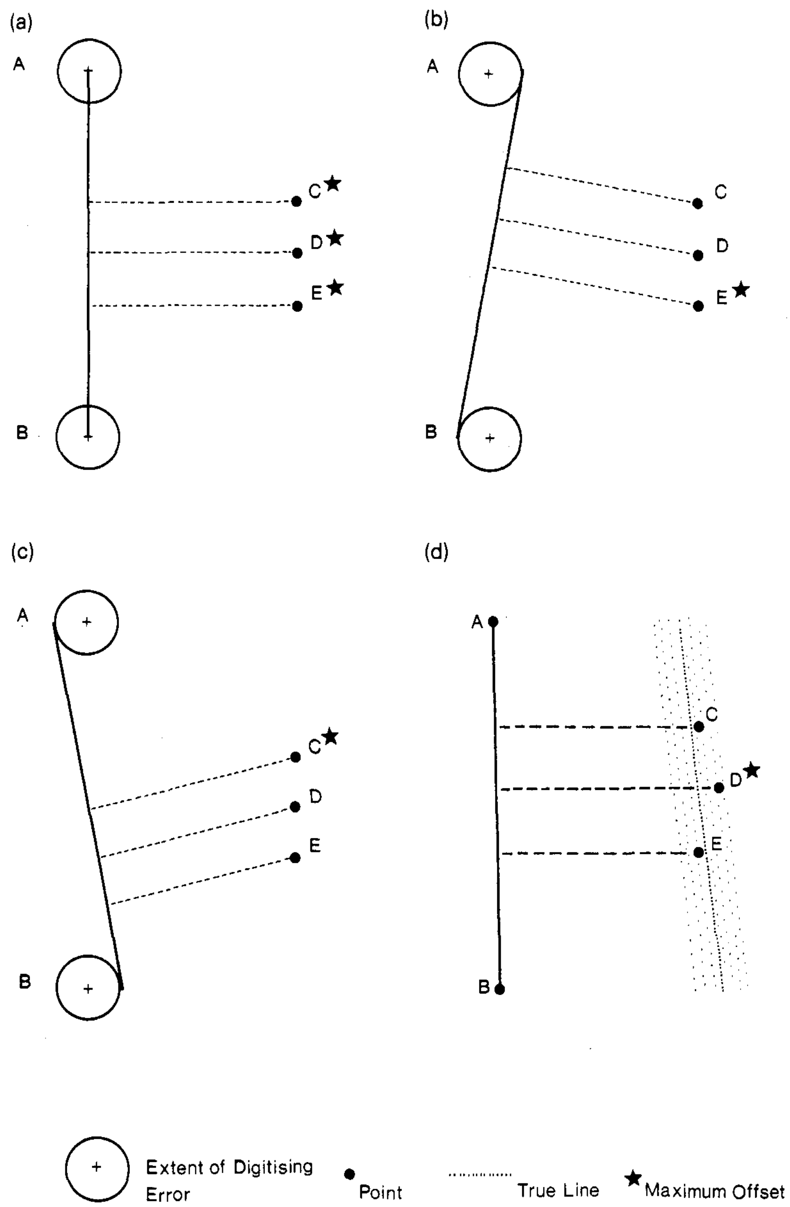

Figure 6. The impact of digitising errors on the maximum offset value.

The presence of digitising errors also implies that the point furthest from the anchor-floater line may also be regarded as distinctive if and only if it does not include other points within its error band as shown in Figure 6d. The difference between the offsets of C, D and E is spurious. As pointed out by Ramer[10], spurious concavities and convexities tend to be introduced during the process of digitising; psychomotor errors tend to cause the operator to oscillate from one side of the line to the other[19]. One of the objectives in line simplification is to remove these aberrations. Yet, the performance of this algorithm is adversely affected by the presence of such errors. Figure 7 shows all points whose offsets are within 5 and 10 metres respectively of the maximum offset (C) in various iterations of the algorithm. These points, particularly those within 5 metres, should be regarded as statistically equidistant from the anchor-floater line.

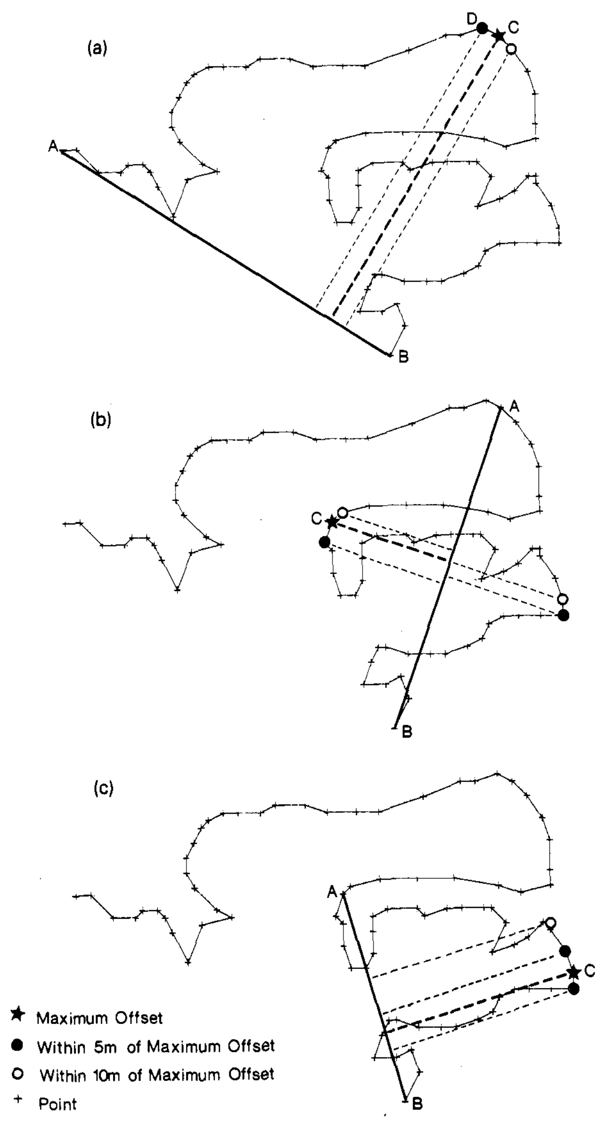

Figure 7. Critical points in the context of digitising errors.

When dealing with line and polygon errors, researchers have tended to measure the goodness of fit of digitised lines with true lines by measuring the total areal displacement of the former. McMaster[2]0 used total areal displacement as an evaluative measure when comparing line simplification algorithms. Could this measure be used to establish whether the deviation between extreme outcomes, obtained by varying the point chosen from the set of equidistant points, is significant? This would involve a consideration of every single permutation of potential selections. We have not pursued this approach for we agree with Muller[8] that total areal displacement is a poor indicator of shape. Cartographic simplifications, like caricatures, are concerned with the preservation of distinctive shapes.

It is impossible to prove conclusively that the presence of digitising errors can be ignored since the results would be dependent upon the selected line configurations. We can however prove the converse, namely that digitising errors impair the performance of this algorithm. For example, in Figure 7a it can be seen that the algorithm selects point C as opposed to D. Since all points are subject to digitising error, point D lying within 5 metres of C is an equally valid but perceptually more significant point. In scale-related generalisations, which conceal the inadequacies of the algorithm to some extent, the rigid use of the maximum offset is acceptable only at the two extreme levels of generalisation. In minimal simplifications, there is a high probability that both points will be included. In very small scale displays, the absolute position of the point is irrelevant. At intermediate levels, the choice could matter, as point C once selected is retained at more detailed levels. The adverse implications of this were discussed elsewhere[4]. It is sufficient to re-state here that the retention of C leads to the non-selection of D even when 40% of points are retained. As a result, the algorithm can exhibit a known weakness of the N'th point method, namely a tendency for cutting perceptually important comers (Figure 8). Also as shown in Figure 7, some candidates communicate very much less visual information and appear to be more dispensable than others. This makes the algorithm particularly unsuitable for scale-independent generalisation. Jenks[21] was justified in being disappointed with the method although he thought that it might have been due to some peculiarity in his version (implementation) of the algorithm; he was probably right in both respects.

The selection of relatively unimportant points on the basis of numerical distances not only prejudices the selection of visually more important ones, but it also means that the algorithm is unnecessarily extravagant - it uses more points than necessary to represent lines. This property of the algorithm was noted by Ramer[10,] who was concerned

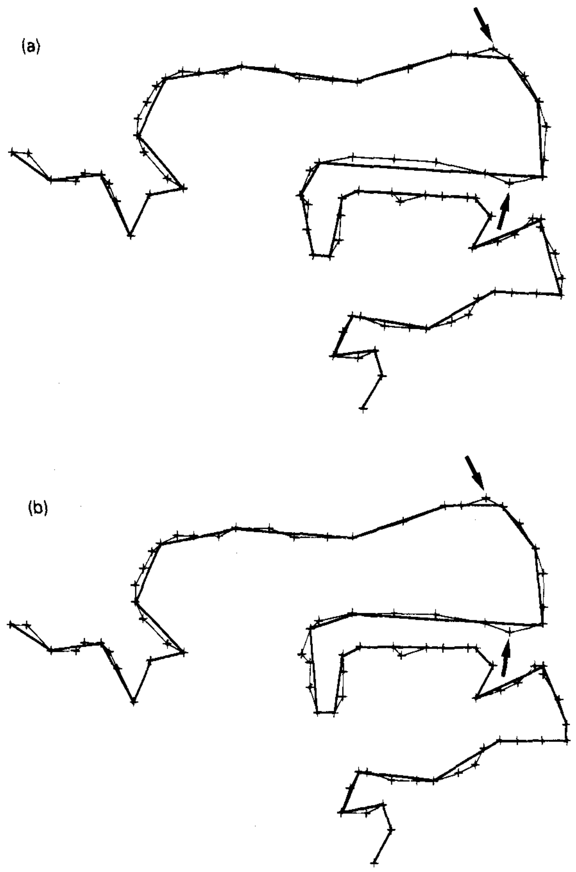

Figure 8. The consequence of spurious accuracy on the shape of the simplified line.

with the approximation of arbitrary 2D curves by polygons. Researchers before him had pursued the ideal objective of representing lines and boundaries by polygons satisfying a given fit criterion, using a minimum number of vertices. Ramer observed that a fit criterion of the maximum distance from the curve to the approximating polygon does not satisfy the ideal objective of locating a minimum number of vertices.

Duda and Hart[9] noted that this algorithm is strongly influenced by individual points and that a single 'wild' point can drastically change the final result. They stressed that many of the heuristics used in image processing and pattern recognition are not dignified by much supporting theory and that they must be used judiciously. They advised that the use of this particular heuristic should be restricted to data that are initially error free. Some researchers (Jones and Abraham',and McMaster[22]) have incorrectly assumed that weeding and/or smoothing remove digitising errors. Weeding cannot make the retained points more accurate; and smoothing can blur the distinctive features of the line.

4. Conclusion

In this paper we used the widely known Douglas-Peucker algorithm to focus attention on the lack of rigour in the expression, interpretation, implementation and evaluation of cartographic software. We also demonstrated that measurement errors can adversely influence the intended effect of such simple algorithms, couched solely in geometric terms. No doubt all generalisations are inaccurate in some respects but this algorithm can never approximate the performance of skilled cartographers. Does this matter? This depends upon the purpose of research. Basic research seeks to develop knowledge and understanding. The discipline of cartography should seek to understand cartographic processes and the cartographer's skills in meaningful and explicit terms so that we have a good grasp of the utility and limitations of our knowledge, techniques and data. The continued promotion of the Douglas-Peucker algorithm by leading researchers stifles innovation and creativity. What is more disconcerting is that this algorithm has already inspired and has become a primitive within secondary spatial analysis and the design of scale-independent databases. Further extensions to the algorithm are also advocated. For example, Goodchild[5] after considering issues relating to the accuracy of spatial databases expressed in a separate section that many of the standard methods for planar spatial analysis, including the Douglas-Peucker line generalisation algorithm, have yet to be adapted to the spherical global context. Those inclined to do so should at least recognise the problems of implementation and resolve them in some rational manner. Even then, the Douglas-Peucker algorithm cannot provide more than a partial and shaky foundation for R & D in line generalisation for it is difficult to envisage how we could standardise the implementation of the algorithm in a meaningful and universally applicable manner. There is also a need to accommodate digitising errors in cartographically meaningful terms.

Finally, we wish to consider the wider implications of this study. Our research has been greatly facilitated by the past practice of detailed publication of research methods; and access by other means not just to algorithms but also their implementations. No doubt those committed to the advancement of knowledge will continue to exchange details of their experimental design and observations (even if they are unable to provide input data provided by research sponsors) so that they can check each other's reasoning and conclusions to mutual benefit. Researchers in computational geometry have pointed out that much spatial software is erected on shaky foundations. Digital cartography builds on computational geometry and computer graphics; Geographical Information Systems in turn embody the academic output of these contributing disciplines within their structures. We hope that this paper has demonstrated in a small way the need for maintaining open and public discussion of the knowledge, techniques and data which underpin the development and use of modem information systems, such as GIS.

Acknowledgements

We are grateful to the Science and Engineering Research Council for the Quota Studentship to Duncan Whyatt. We also wish to thank Phil Wade of the Cartographic Information Systems Research Group (CISRG) at the University of Hull for his implementation of the Douglas-Peucker algorithm and Mike Blakemore of the Geography Department at the University of Durham for the source of the original implementation by Douglas. As a part of his Final Year project in the Department of Computer Science at the University of Hull, Kurt Dowson developed a program which enabled us to visualise the detailed execution of the Douglas-Peucker algorithm. We also wish to thank Drs Graham Kirby and Mike Turner of the CISRG and the referees for their critical comments on aspects of this paper, and our colleague, Dominic Varley, for drawing our attention to a recent electronic bulletin board debate on the accuracy of floating point calculations. Some of the material in this publication has been produced using data or other material relating to the digitised boundary information and these data or other material remain the property and copyright of the Crown.

References

1. Forrest, A.R., "Computational Geometry in Practice," pp. 707-724 in *Fundamental Algorithms for Computer Graphics,* ed. R.A. Earnshaw, NATO ASI Series F17, Springer-Verlag, Berlin (1985).
2. Douglas, D.H. and Peucker, T.K., "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature," *Canadian Cartographer* **10(2),** pp. 112-122 (1973).
3. McMaster, R.B., "Automated Line Generalization," *Cartographica* 24(2), pp. 74-1 11 (1987).
4. Visvalingam, M. and Whyatt, J.D., "The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation Through Visualisation," *Computer Graphics Forum* 9(3), pp. 213-228 (1990).
5. Goodchild, M.F., "The Issue of Accuracy in Global Databases," pp. 31-48 in *Building Databases for Global Science,* ed. H. Mounsey, Taylor & Francis, Basingstoke (1988).
6. Buttenfield, B.P., "Digital Definitions of Scale Dependent Line Structure," pp. 497-507 in *Proc. Auto Carto London (ICA),* ed. M. Blakemore (1986).
7. Jones, C.B. and Abraham, I.M., "Line Generalization in a Global Cartographic Database," *Cartographica* 24(3), pp. 32-45 (1987).
8. Muller, J.C., "Fractal and Automated Line Generalization," *cartographic Journal* 24, pp. 27-34 (1987).
9. Duda, R.O. and Hart, P.E.. pp. 337-339 in *Pattern Classification and Scene Analysis,* Wiley, NY (1973).
10. Ramer, U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Computer Graphics and Image Processing* 1, pp. 244-256 (1972).
11. Douglas, D.H., *Collected Algorithms,* Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, Massachusetts (1975).
12. White, E.R., *Perceptual Evaluation of Line Generalization Algorithms,* Unpublished Masters Thesis, University of Oklahoma (1983).
13. Whyatt, J.D. and Wade, P.R., "The Douglas-Peucker Line Simplification Algorithm," *Society of University Cartographers Bulletin* 22(1),pp. 17-25 (1988).
14. Waugh, T.C. and McCalden, J., *GIMMS Reference Manual (4.5),* Gimms Ltd, 30 Keir Street, Edinburgh EH3 9E4 (1983).
15. Peucker, T. K., "A Theory of the Cartographic Line," *Proc. Auto Carto* **2,** pp. 508-518, Reston, Virgina (1975).
16. Huggins, S.R., *Programming in* C, transputer@prg.oxford.ac.uk mailing list, Software Development Group, INMOS Ltd, 1000 Aztec West, Almondsbury, Bristol BS12 4SQ, April 26 (1991).
17. Blakemore, M., "Generalisation and Error in Spatial Databases," *Cartographica* 21(2&3), pp. 131-139 (1984).
18. Perkal, J., *An Attempt at Objective Generalization,* Translated by W. Jackowski from Julian Perkal, Proba obiektywnej generalizacji. Geodezia es Kartografia, Tom VII, Zeszyt 2, 1958, 130-142. In J. Nystuen (ed) Michigan Inter-University Community of Mathematical Geographers, Discussion Paper 10. Ann Arbor, University of Michigan (1966).
19. Jenks, G.F., "Lines, Computers, and Human Frailties," *Annals of the Association of American Geographers* 71(1), pp. 1-10 (1981).
20. McMaster, R.B., "The Geometric Properties of Numerical Generalization," *Geographical Analysis* 19(4),pp. 330-346 (1987).
21. Jenks, G.F., "Thoughts on Line Generalisation," pp. 209-220 in *Auto Carto 4: Proc. International Symposium on Cartography and Computing: Applications in Health and Environment (American Congress on Surveying and Mapping),* ed. R. T. Aangeenbrug (1979).
22. McMaster, R.B., "The Integration of Simplification and Smoothing Algorithms in Line Generalisation," *Cartographica* **26(1),** pp. 101-121 (1989).