Discussion Paper Series Editors: Dr M Visvalingam

C.I.S.R.G. DISCUSSION PAPER 16

A Computer Science Perspective on the Bendsimplification Algorithm

by

M Visvalingam and S P Herbert

May 1998

CISRG Discussion Papers

1. Wade, P., Visvalingam, M. and Kirby, G. H. (1986)
   From Line Geometry to Area Topology, 48 pp.
2. Visvalingam, M. (1987)
   Problems in the Design and Implementation of a GKS-based User Interface for a Graphical Information System, 18 pp.
3. Visvalingam, M. (1988)
   Standardising Basic Spatial Units: Problems and Prospects, 19 pp.
4. Whyatt, J. D. and Wade, P. R. (1988)
   The Douglas-Peucker Algorithm for Line Simplification: an introduction with programs, 17 pp.
5. Visvalingam, M. (1988)
   Cartography, Geographical Information Systems and Maps in Perspective, 16 pp.
6. Visvalingam, M. and Whyatt, J. D. (1989)
   The Douglas-Peucker Algorithm for Line Simplification: re-evaluation through visualisation, 35 pp.
7. Visvalingam, M. (1989)
   Trends and Concerns in Digital Cartography, 37 pp.
8. Visvalingam, M. and Whyatt, J. D. (1990)
   Cartographic Algorithms : problems of implementation and evaluation and the impact of digitising errors, 24 pp.
9. Visvalingam, M. (1991)
   Visualisation, VISC and Scientific Insight, 21 pp.
10. Visvalingam, M. and Whyatt, J. D. (1992)
    Line Generalisation by Repeated Elimination of the Smallest Area, 16 pp.
11. Varley, D. A. and Visvalingam, M. (1993)
    Area Topology for Road Extraction and Topographic Data Validation, 34 pp.
12. Visvalingam, M., Varley, D. A., and Wright, C. P. (1993)
    Global dimensions for the recognition of prototypical urban roads in large-scale vector topographic maps, 20 pp.
13. Visvalingam, M., Williamson, P.J. (1993)
    Generalising Roads on Large-Scale Maps: a comparison of two algorithms, 21 pp
14. Visvalingam, M. (1996)
    Approximation, Generalisation and Deconstruction of Planar Curves, 13 pp.
15. Visvalingam, M. and Dowson, K. (1997)
    Algorithms for Sketching Surfaces, 19 pp.
16. Visvalingam, M. and Herbert, S. P. (1998)
    A Computer Science Perspective on Bendsimplification, 17 pp

Special Issues

1. Visvalingam, M. (ed) (1986)
   Research based on Ordnance Survey Small-Scales Digital Data Proceedings of a meeting held as part of the 1986 Annual Conference of the Institute of British Geographers (Reading Jan 1986), **sponsored by the Ordnance Survey,** 79 pp.
2. Visvalingam, M. and Kirby, G. H. (1987)
   Director of Research and Development based on Ordnance Survey Small Scales Digital Date, **sponsored by the Ordnance Survey**, 38 pp.
3. Visvalingam, M. and Sekouris, N. M. (1989)
   Management of Digital Map Data Using a Relational Database Model, **sponsored by the Ordnance Survey**, 50 pp.

CONTENTS

ABSTRACT

This paper provides a conceptual and empirical evaluation of the Bendsimplify algorithm presented by Wang and Muller (1998) and identifies some problems with its Arc/Info implementation. Experimental results suggest that the algorithm may not be as widely applicable as the much simpler geometric filters, such as the Douglas-Peucker or Visvalingam algorithms. The paper questions the value of over-coupling model- and image-oriented generalisation processes within a black-box system. It suggests the type of parameters, which could enhance the utility and usability of the Bendsimplify option within the Arc/Info (and perhaps also within the ArcView) environment and concludes with some pointers for further research. This investigation was undertaken to establish whether Bendsimplification is more useful for line segmentation than Visvalingam's algorithm. The evidence suggests that it is not so.

KEYWORDS: line generalisation; bendsimplification; Arc/Info; Visvalingam's algorithm

1. Introduction

Visvalingam and Williamson (1995) noted that the iterative point elimination algorithm, proposed by Visvalingam, offers some scope for automatic segmentation of in-line features for knowledge-based generalisation of lines. Visvalingam's iterative removal of triangular features, subtended by points and their neighbours, results in the progressive elimination of scale-related features (Whyatt, 1991; Visvalingam and Whyatt, 1993). The bendsimplify algorithm (Wang and Muller, 1998) extends this concept to involve the iterative elimination of convex and concave bends. Bends are defined as sections of curves between two inflection points. Since bends were likely to be better indicators of features than triangles, Herbert (1998) explored the Bendsimplify option within Arc/Info 7.1.1 within a project concerned with line segmentation. The results were unexpected and prompted further evaluation of this new algorithm.

In this paper, the word - ***bendsimplification*** - is used to refer to the algorithm described by Wang and Muller (1998), to differentiate it from the ***Bendsimplify*** option within Arc/Info 7.1.1. It is not the aim of this paper to provide an exhaustive evaluation of this particular implementation of bendsimplification, which may be limited by the host software environment. Its objective is to identify conceptual problems and to suggest further algorithm-oriented research.

2. Does the "analysis of shape characteristics" reveal line structure ?

This study was prompted by the idea that one process within bendsimplification, namely iterative bend elimination (Wang, 1995), appeared to be useful for revealing line structure. However, the paper by Wang and Muller (1998) was not unequivocal about the value of bendsimplification in structure analysis. Hence, one objective of the study was to resolve possible misinterpretations in the reading of their text. They initially note that "Bends are hidden behind the x-y coordinates of the line feature and reflect a line structure which must be computationally revealed. Therefore, *the major task is to write a program capable of understanding the structure of spatial information... i.e. detecting bends and computing their attributes on which generalization decisions can be made.*" (p 4, our emphasis) and that "Most existing line generalization solutions are based on geometric processing without previous shape analysis ... " (p 5). Although bendsimplification is still in the realms of geometric processing, the above statements seem to imply that it is guided by the structuring which results from shape analysis (as suggested by the title of the paper).

Later, they qualify these statements by stating that their experimental system "can only recognize bends, evaluate such attributes as area, shape, length of bend, span of bend, or length of the baseline, and assess the context based on these attributes, such as bend similarity and isolation. There are still other line characteristics which are ignored in the current system. For example, irregular coastlines often contain deep and branched bays and this kind of line characteristics has an even higher level of information since each of them consists of numerous bends " (p 14). However, their Lake Shoreline example of bendsimplification (their Figure 12.5) shows such convoluted and branched lines, consisting of features within features. So, the present empirical tests were undertaken to get a better understanding of the authors' statements on this matter.

When shown some results output using the Arc/Info Bendsimplify option, Wang and Lee (personal communication, Feb 1998) clarified that the concept of bends referred only to simple bends and that it did not include deep bays with branches. Since these results raise other debatable issues, they are presented and discussed here.

3. Criteria for Evaluation

Unlike the Douglas and Peucker (1973) and the Visvalingam (Visvalingam and Whyatt, 1993) algorithms, bendsimplification is not concerned with minimising the number of points used for representing curves. It is more concerned with articulating four guidelines for manual generalisation as provided by the Swiss Society of Cartography in 1997 (see Wang and Muller, 1998). These 'rules' include the retention of isolated small bends, the combination of similar adjacent bends, and the exaggeration of these bends. Thus, the usual process of comparing the output obtained by using an equivalent number of points is not strictly applicable but is, nevertheless, useful.

Given the aims of bendsimplification, the mathematical measures of evaluation, proposed by McMaster (1987), are also inappropriate. As a result, this evaluation is largely based on visual judgement based on perceptual criteria. Fractal curves have proved to be good test lines for evaluating the geometric properties of line generalisation algorithms (Visvalingam, 1996). With the quadric Koch island, Visvalingam and Brown (1999) found that the complex symmetry of these curves was better preserved by Visvalingam's algorithm, compared with the Douglas-Peucker algorithm. This was especially the case when effective area was used as the measure of significance driving the iterative elimination. The quadric Koch island was therefore included in the empirical evaluation of bendsimplification.

4. Bendsimplification within Arc/Info 7.1.1

The following problems were encountered when evaluating the Bendsimplify option within Arc/Info:

1. *There is an upper limit of 500 points per arc* (Dan Lee and Zeshen Wang, ESRI, personal communication, February 1998). Since Wang and Muller (1998) were able to process much longer lines, it was suggested that there were problems in the integration of the Bendsimplify option within the Arc/Info environment. The system appears to segment long lines into 500-point node-connected arcs. Since these pseudo-nodes are not moved or eliminated, the output resulting from long lines was far from satisfactory. However, further investigation with other lines suggests that there

may be other, more fundamental, problems as noted in this paper.

2. *There is also a lack of user access to the metric information used by the software to drive the generalisation* of both the Douglas-Peucker and Bendsimplify options to the ***generalize*** command in Arc/Info. The user, therefore, is obliged to select tolerances through a tedious process of trial-and-error driven by guess-estimates, as recommended by the on-line help system. Wade's implementation of the Douglas-Peucker algorithm (reported in Whyatt and Wade, 1988) associates a distance value with each vertex of the line during a one-off tagging phase so that the line may be dynamically filtered at display time. The CISRG implementation of Visvalingam's algorithm not only associates the measure of significance used (such as, effective area, distance or base length), but also outputs a rank value to ease the selection of tolerance values.

Therefore, despite diligence on our part, we are uncertain as to whether the tolerances used are the best ones. Nevertheless, the results are noteworthy, particularly since they suggest future research and development, which could be undertaken both within and outside ESRI.

5. Observations

5.1 Large-scale data for roads

Figure 1 shows sample output from the Bendsimplify option when provided with the 1:1250 large-scale data for a road boundary, consisting of 270 points, which was previously used by Visvalingam and Williamson (1995). The results speak for themselves. Like Visvalingam's algorithm, Bendsimplify also progressively eliminates scale-related forms. Initially, it produces results that are similar to those produced by Visvalingam's algorithm but using many more points. Visvalingam and Williamson (1995) produced a better result than that shown in Figure 1a with just 33 points; and a much more appropriate result corresponding to Figure 1c with just 16 points. Visvalingam's algorithm defines the broad shape of the main road boundary before bringing in the branch road; Bendsimplify eliminates the former (Figure 1d) before the latter (Figure 1e). The exaggeration of the branch roads is not appropriate here (Figures 1a - d); this is not surprising since roads are not bends in coastlines. Even after the elimination of most features, Bendsimplify retained a large number of points contrary to Wang and Muller's (1998) belief that large reduction rates are possible with bendsimplification. The results show that, unlike the Douglas-Peucker and Visvalingam algorithms, *Bendsimplification is not a general-purpose algorithm.*

5.2 Coastlines

Arc/Info's Bendsimplify option was then tested using the coastline of Carmarthen Bay, previously used by Visvalingam and Whyatt (1993). Whyatt (personal communication, 1998) provided some

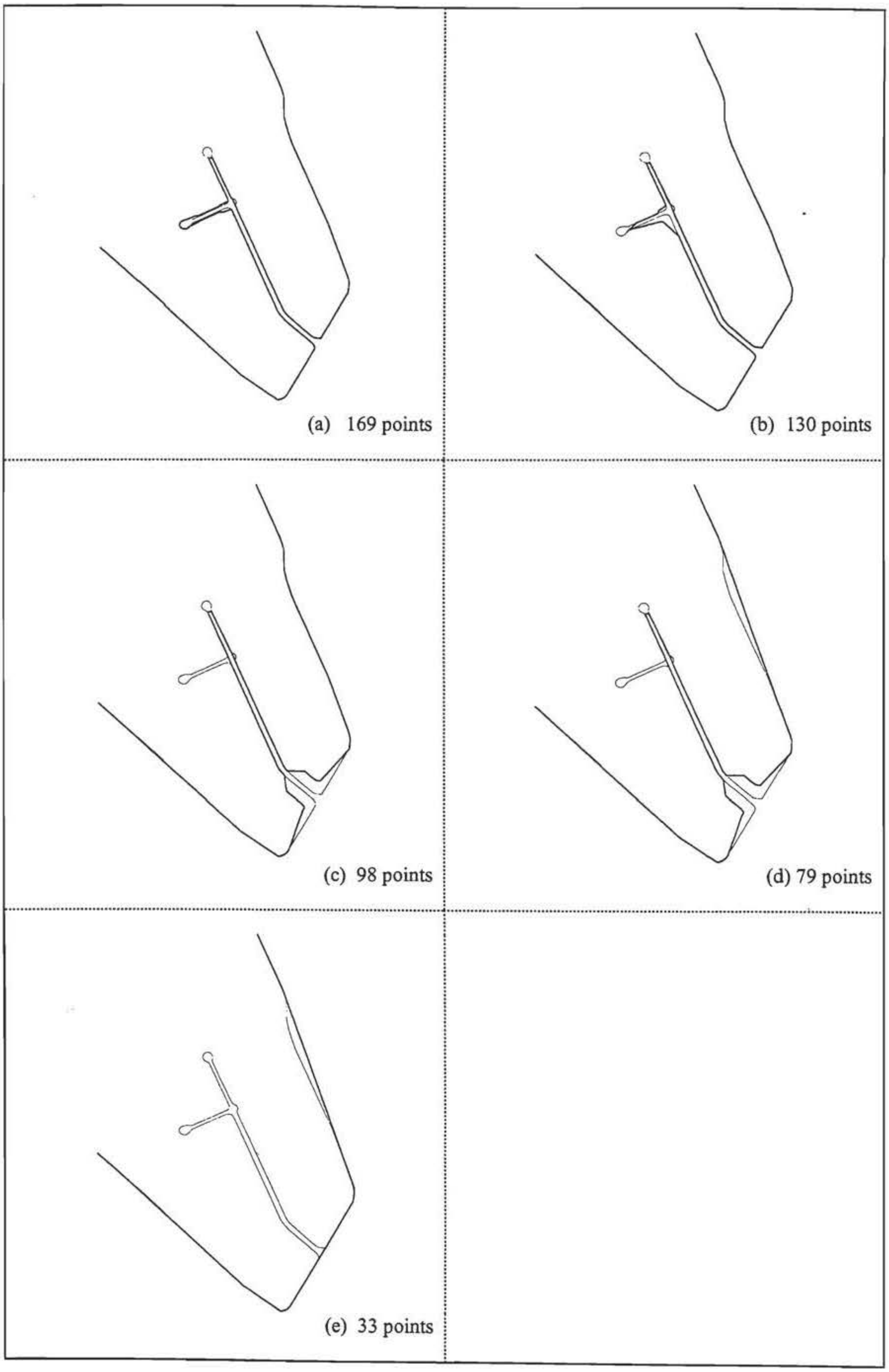


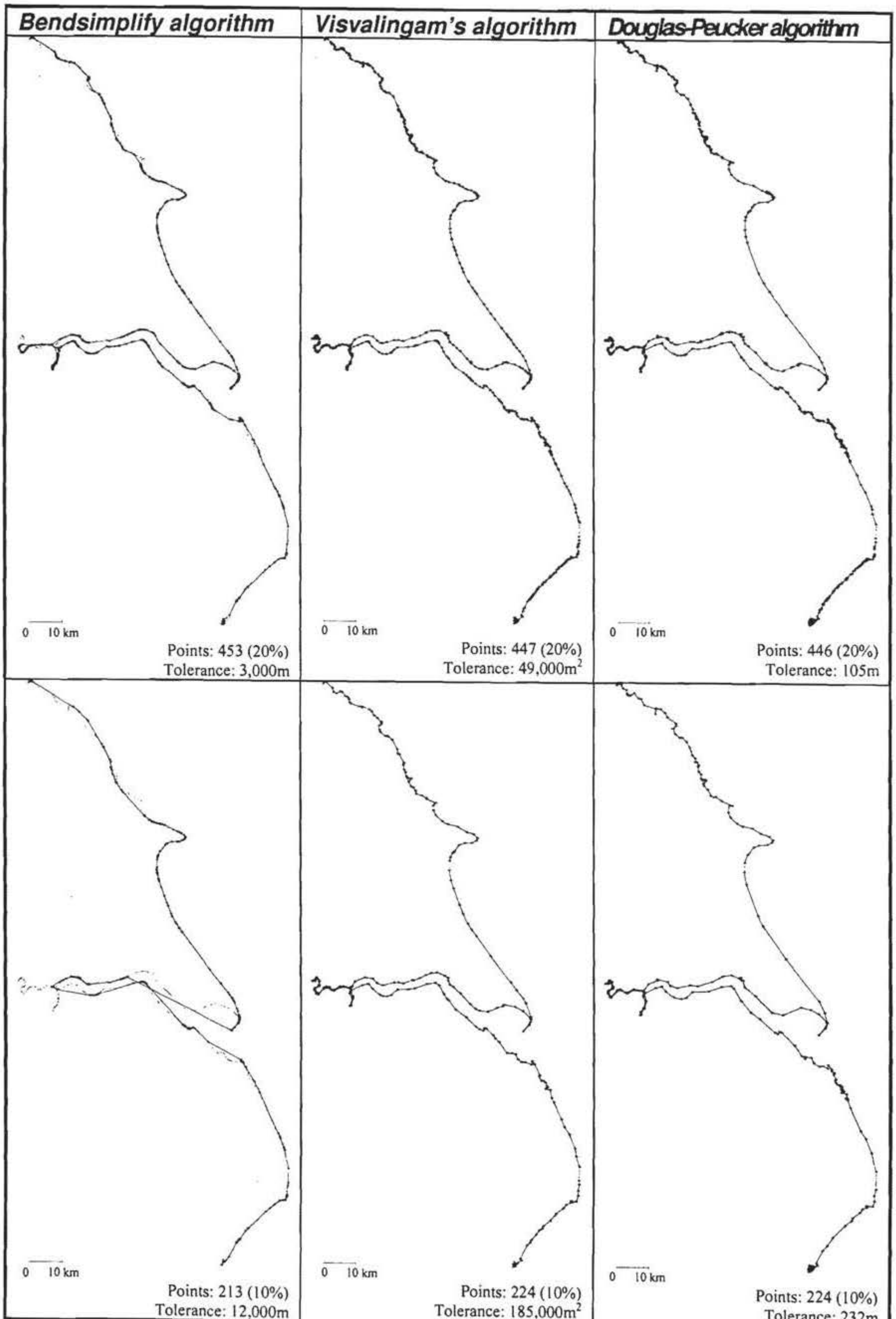
Figure 1 : **Bendsimplification of a road boundary, captured at 1 : 1250 scale**
Data source: Ordnance Survey, Crown Copyright reserved

Bendsimplify results based on this test data. The results did not compare well with those produced by Visvalingam's algorithm and those with 25% or less points were especially strange. Sections of coastline continued to be depicted in great detail but were connected by straight lines, which truncated large sections of the coastline in inappropriate places. Wang (ESRI, personal communication, 1998) found that the end points of these straight-line sections were located on the spurious nodes, inserted by Arc/Info. Dan Lee and Zeshen Wang, therefore, believed that this test line, which consists of 1583 points, was inappropriate and that the data had been subjected to higher levels of generalisation than the algorithm was designed for. Whyatt's results are therefore excluded from this report although Visvalingam's algorithm produces quite effective and appropriate caricatures of coastlines with less than 19 points (less than 2%), and even just 4 points (see Visvalingam and Whyatt, 1993).

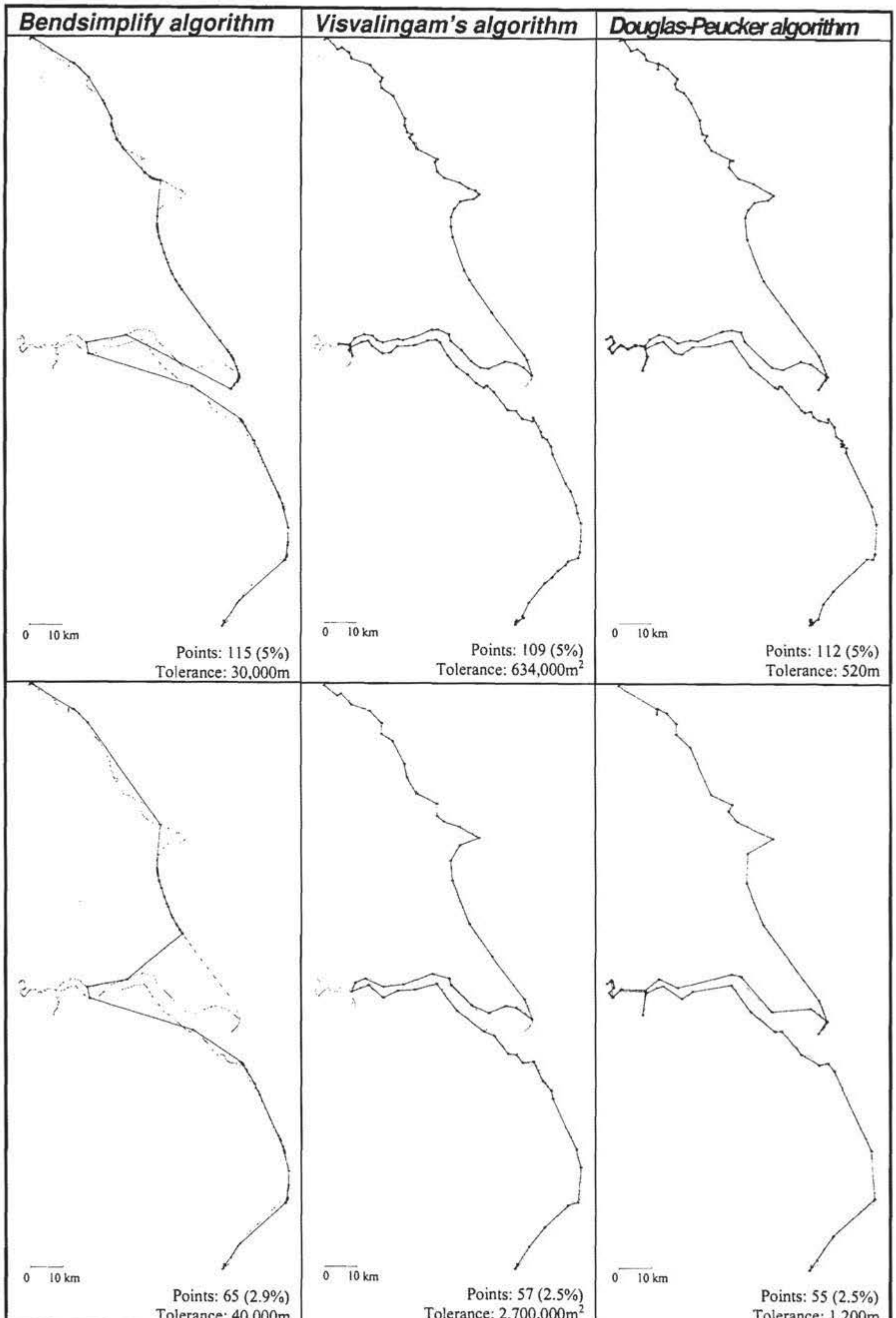
A more detailed investigation was undertaken, by the authors of this paper, using the 1:50000 coastline of Humberside. Figure 2 shows that the Douglas-Peucker and Visvalingam algorithms can filter down to even 2.5% of points, although the results of the former are adversely affected by the extreme points on the River Ouse. The Bendsimplification algorithm produces very unbalanced results. This coastline does not have a complex hierarchy of features within features as in Carmarthen Bay, but it features two meandering rivers. Visvalingam and Whyatt (1993) demonstrated that Visvalingam's algorithm could achieve good minimal simplification of the head of the estuary using less than 25% of points; With 5% and 2.5% of points it is possible to achieve the style of depictions provided in atlas maps (see Figure 2). A cartographer would no doubt have exaggerated the mouth of the rivers in ***visual*** mapping; but this introduces unnecessary complications for line segmentation, which is a part of data modelling - i.e. ***digital*** mapping (Visvalingam, 1989). The programme of research within the CISRG assumes that database-oriented digital mapping could facilitate image-oriented interactive visual mapping.

Since Arc's segmentation of the line into 500-point sections may have affected the simplification process to some extent, a 500 point section of the line at the head of the Humber Estuary was studied to remove Arc intervention. Also, by its very name, Bendsimplification, is said to be designed for minimal simplification - not caricatural generalisation. Hence, the comparisons made in Figure 2 may appear irrelevant to some. However, Figure 3 shows that even if we reduce the tolerance to retain over 50% of points, the output is still unbalanced. Figure 3a, with 58.2% of points provides the sort of minimal simplification provided by Visvalingam's algorithm with just 22% of points (see Visvalingam and Whyatt, 1993). Figure 3b, with 57.4% of points shows line intersections, and a curious hooked feature, resulting from differential displacement of the two banks of the furthest meander. The problem of differential treatment of the two banks is especially evident in Figure 3c with 35% of the points where it results in a lake, attached to the head of the estuary by two straight lines. Thus, the unbalanced results observed at gross levels of filtering are also present at the 35% level of filtering, which generally supports excellent minimal simplification with especially the Douglas-Peucker algorithm. Figure 3c also demonstrates another consequence of using a complex algorithm, such as Bendsimplification.

Figure 2 : Generalisations of the 1 : 50000 coastline of Humberside, UK
Data source: Ordnance Survey, Crown Copyright reserved



Continued / . . .



Bendsimplify algorithm
Visvalingam's algorithm
Douglas-Peucker algorithm
0 10 km
Points: 115 (5%)
Tolerance: 30,000m
0 10 km
Points: 109 (5%)
Tolerance: 634,000m2
0 10 km
Points: 112 (5%)
Tolerance: 520m
0 10 km
Points: 65 (2.9%)
Tolerance: 40,000m
0 10 km
Points: 57 (2.5%)
Tolerance: 2,700,000m2
0 10 km
Points: 55 (2.5%)
Tolerance: 1,200m

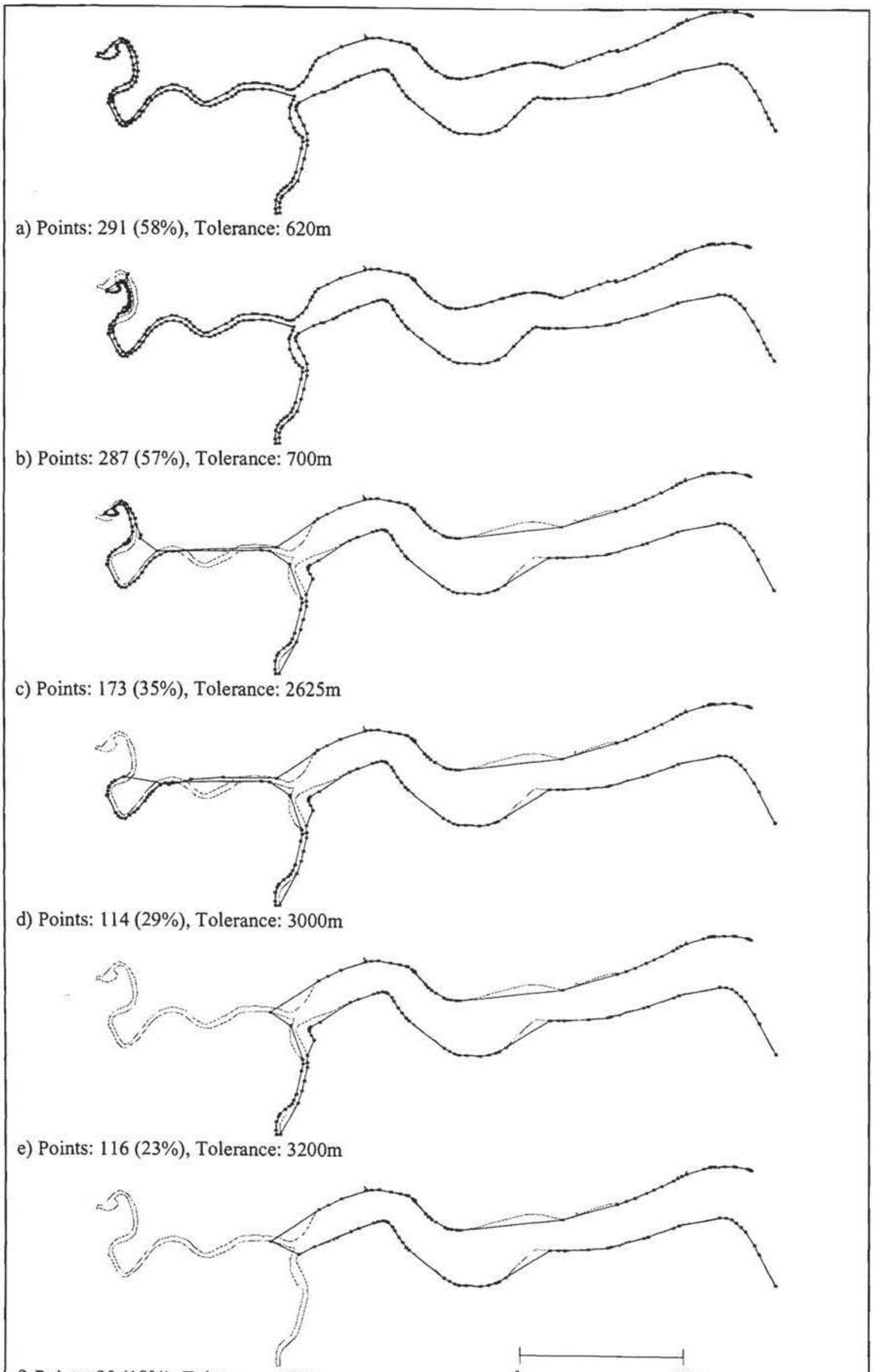


Figure 3: Bendsimplification of the Humber Estuary - enlarged

The use of a larger tolerance has made the algorithm reuse a feature that had been discarded when using a lower tolerance. This is strange given that straight lines have replaced a long stretch of the river, as noted earlier. Note also that the mouths of the rivers have become widened and distorted in Figure 3c, as the roads had been in Figure 1. Also, the salient character or the coastline is not retained since the longer river is eliminated, before the shorter one (Figure 3e). In Figure 3f, only one of the points is appropriate for segmenting the head of the estuary. Visvalingam's algorithm with 5% of points (Figure 2) provides a more suitable filtering tool for research into line segmentation. It is inevitable that the output of line-based algorithms is likely to cross. Whereas this is only an issue with about 10% of the Humberside points in the case of of Visvalingam's algorithm, Figure 3b (with over 50% of points) is already manifesting Bendsimplification's inability to handle tight bends. Given the clues (for segmentation) in the output of Visvalingam's algorithm, line crossings are no longer problems but provide further clues for selecting scale-related styles of generalisation that would be appropriate for the segmented parts of lines (see Visvalingam and Whyatt, 1993). The results of Bendsimplification, therefore, suggest that Visvalingam's algorithm may be more useful for deriving an intelligent approach to generalisation.

5.3 The quadric Koch island

The Bendsimplify option was then tested using fractals. The rectangular Koch island was chosen since it makes it easier to investigate the reasons for Bendsimplify's retention of a large number of points, noted earlier. Fractal curves are generated by repeatedly applying a transformed version of a generator pattern to the edges in a curve. The presence, by definition, of scale-related features within features makes fractals useful for testing line generalisation algorithms. Mandelbrot (1983) coined the term *teragons* to refer to specific generations of fractal curves. Visvalingam (1996) coined the term *decogon* to denote a decorative, rather than a meaningful, pattern obtained by deconstruction. She differentiated deconstruction from the complementary processes of approximation (or minimal simplification) and generalisation respectively. Whereas manual generalisation is largely knowledge-based and is biased towards the abstraction of meaningful patterns from given data, deconstruction is an entirely mechanical process aimed at revealing unforeseen patterns and structures in data. Visvalingam and Brown (1999) used the Douglas-Peucker and Visvalingam algorithms as deconstructors to study their geometric properties and to note the symmetry elements that were preserved.

Given the 500-point limit on Arc/Info arcs, only the first two generations of the quadric Koch island are used as test lines in this paper. The level 1 teragon was initially considered since it only has simple 'bends' with no hierarchy of nested bends.

Figure 4, adapted from Visvalingam and Brown (1999), shows various decogons obtained by filtering the first generation teragon of the quadric Koch curve, which consists of 32 points. Given this 32 point level-1 teragon, Bendsimplify returns either the input line or the square initiator, which is identical that

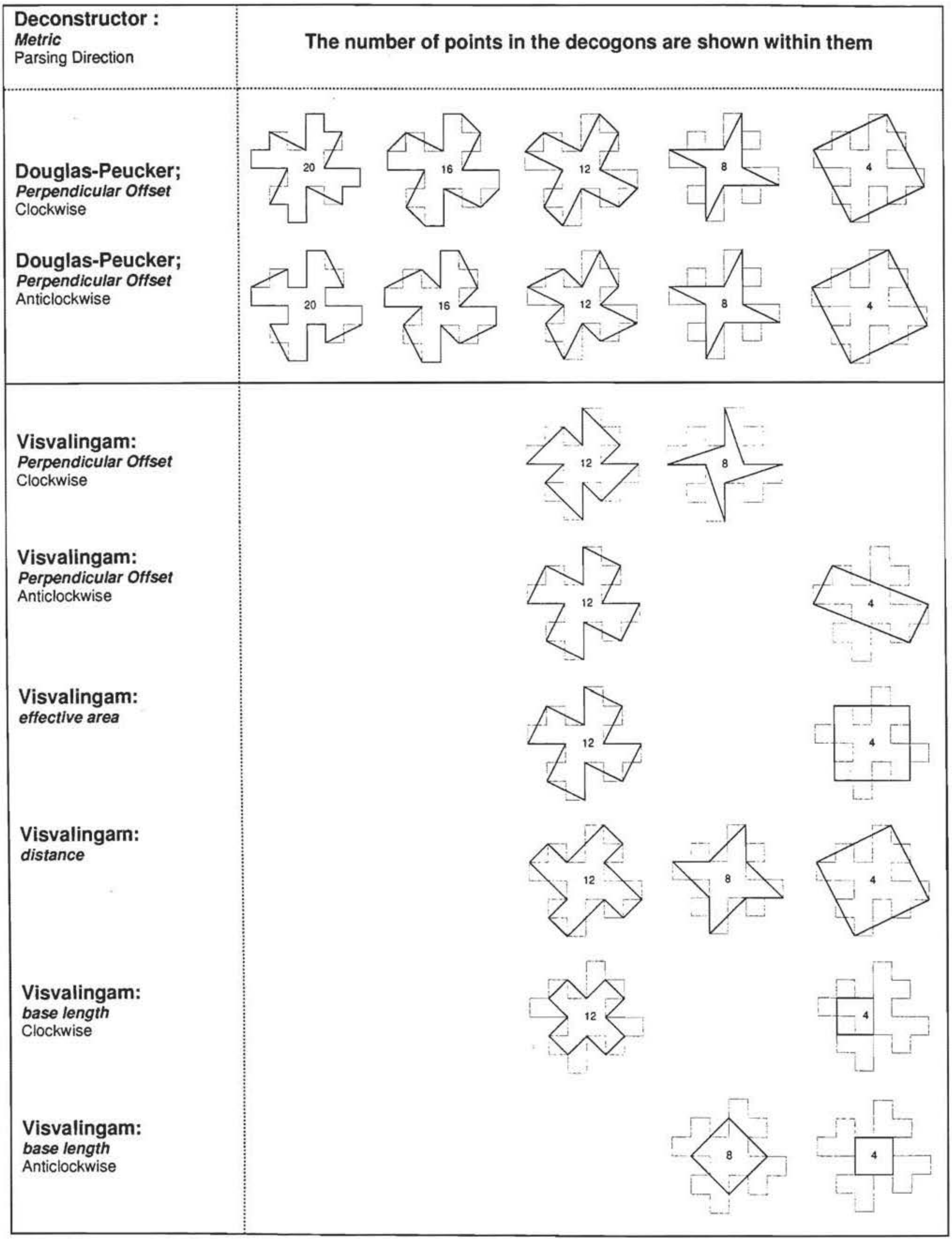


Figure 4 : Various deconstructions of the level 1 quadric Koch teragon
(Adapted from : Visvalingam and Brown, 1999)

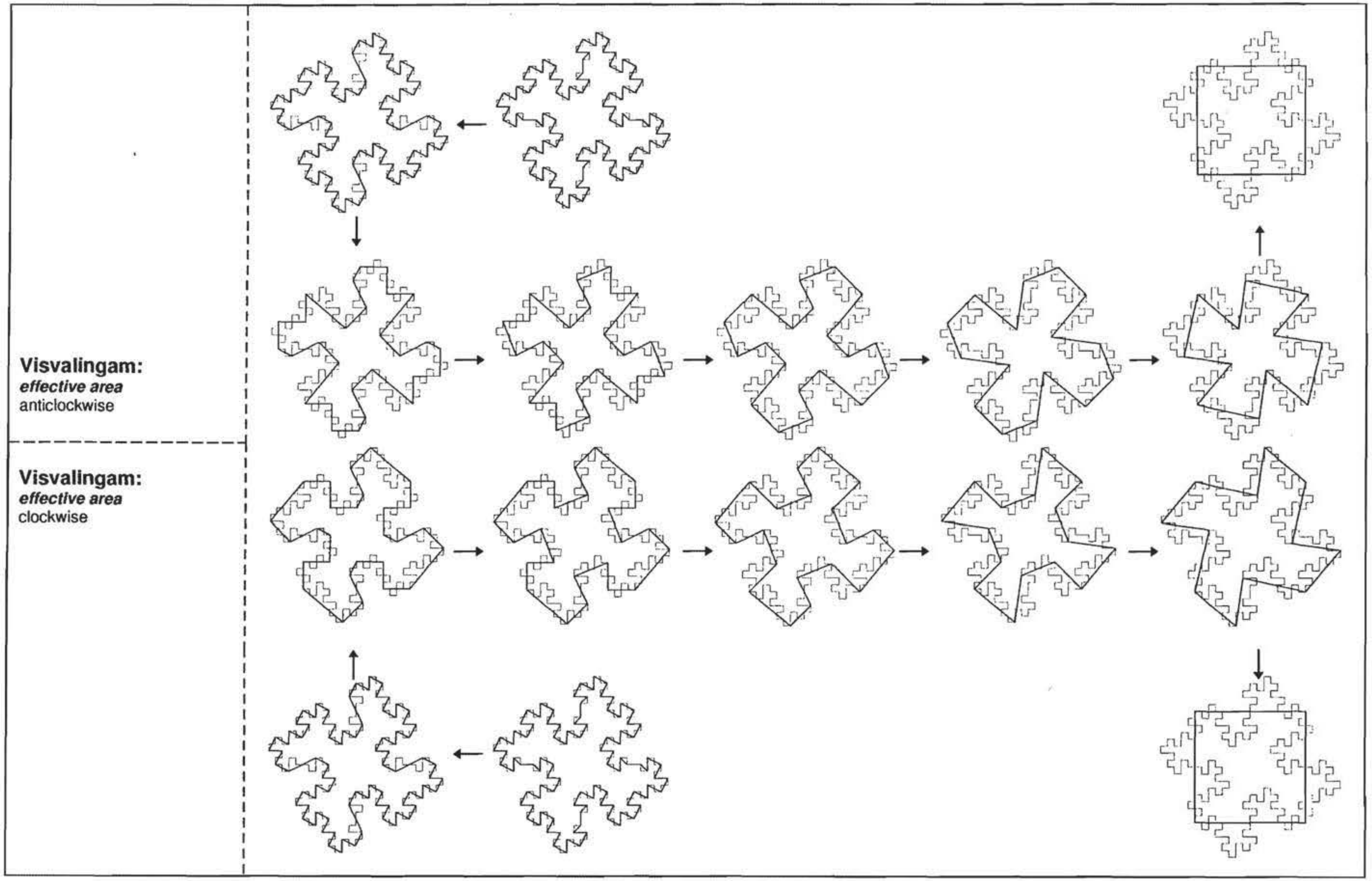
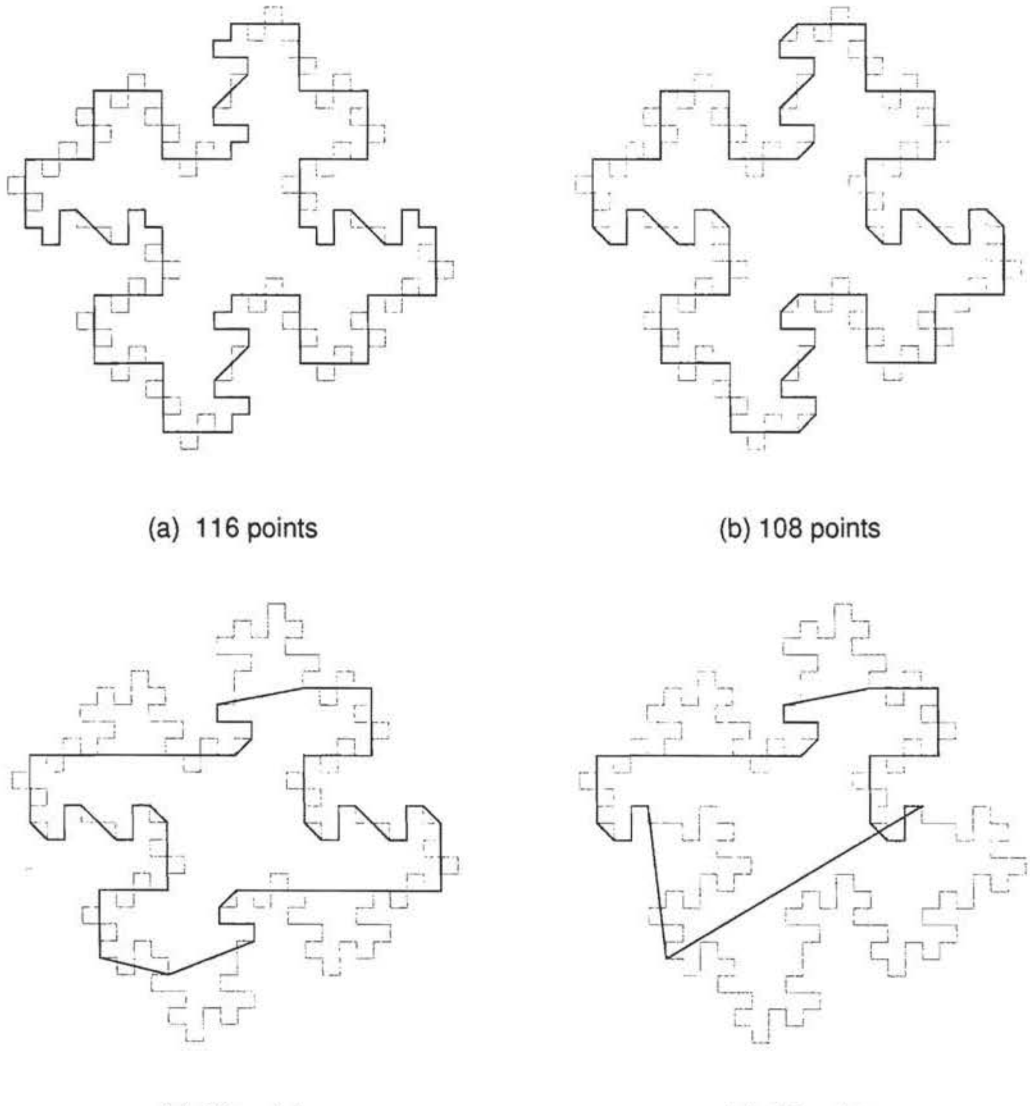


Figure 5 : Deconstruction of the level 2 quadric Koch teragon by Visvalingam's algorithm using effective area.
Source : Visvalingam and Brown (1999)

produced by Visvalingam's algorithm with effective area in Figure 4. However, it retains 11 points to represent the square even though 6 of the retained points lie on straight lines and are therefore redundant. Increasing the tolerance reduces the teragon to the two - start and end - points of the curve. This provides undeniable evidence that Bendsimplify was not achieving the large reduction rates anticipated by Wang and Muller (1998).



(a) 116 points

(b) 108 points

(c) 71 points

(d) 38 points

Figure 6 : Bendsimplification of the level 2 quadric Koch teragon

Figures 5 and 6 show the results obtained from the level-2 teragon (consisting of just 256 points) using Visvalingam's algorithm and the Bendsimplify option respectively. A detailed analysis of Figure 5 is provided elsewhere (Visvalingam and Brown,1999); it is sufficient to note here that the 4-fold symmetry is maintained throughout progressive deconstruction. Figure 6a shows the first reduction of the line by Bendsimplify. Whereas Visvalingam's algorithm only uncovers the level 1 teragon from the level 3 teragon upwards (not shown here), and not from the level 2 teragon, Bendsimplify abstracts a reasonable depiction of the level 1 teragon from the level 2 teragon. It preserves the 4-fold symmetry in Figures 6a and 6b. True to its objective, it has amalgamated bends in parts of the line and has retained more detailed elements in other parts, giving an intentionally unbalanced result. It looks as if there has been some degree of bend elimination and/or bend amalgamation but the exaggeration operator does not appear to have been fired. The decogon in Figure 6b, in particular, is quite pleasing.

On further reduction, the 4-fold symmetry is lost in Figures 6c and 6d. All four arms of the teragon were not treated similarly; identical bend configurations were processed differently. Iterative bend elimination on its own should have retained the four-fold symmetry. So, it is likely that the loss of symmetry arises from bend amalgamation since there is no evidence here of the type of exaggeration seen in Figures 1-3. Figure 6d was unexpected and the shape with crossing lines are similar to problems encountered with coastline data which indicates that the problems encountered with the coastline are not entirely due to the 500 point maximum limit on arcs. Visvalingam (1996) noted that many empirical derivations of fractal dimensions used inappropriate algorithms and that some of the published figures suggest that the implementation had not taken account of rounding errors in calculation. By definition, fractals tend to have many identical elements. Rounding errors in the calculation of heuristics create unbalanced results. Digitising errors are much larger than such rounding errors and influence the results of some algorithms (Visvalingam and Whyatt, 1991); it could be that the implementation of Bendsimplification is not taking account of this problem.

The results obtained using Arc/Info's Bendsimplify option, taken as a whole, are worse than those produced by the Douglas-Peucker algorithm (see Visvalingam and Brown, 1999). Although the latter has a propensity to ***create*** spikes where none exist and produce an unbalanced generalisation that is now well known, it does preserve the 4-fold symmetry down to four points. Compared with the 38-point Figure 6d, the penultimate cross-shapes in Figure 5 only consist of 12 points. Bendsimplify does not cull redundant points.

6. Discussion and Future Work

Given the caveats about the side effects generated by the host environment, it is difficult to establish the degree of correspondence between the Bendsimplify option and the Bendsimplification algorithm. Wang and Muller (1998) did state that the four rules on which they have based their algorithm are not exhaustive and that, as yet, their experimental system does not encapsulate even these four rules fully. They concluded with "There is also a need for additional cartographic rules for line structure recognition to enable more sophisticated operations" (p 14). However, the results presented in this paper suggest that the bendsimplification approach may not be widely applicable. The recognition, combination and representation of bends seem to make some assumptions about bends, which are not appropriate for road outlines, coastlines or fractals.

The Bendsimplification approach is distinguished by its "Integration of numerous operations (e.g. elimination, exaggeration and combination) into a single program" (p 13). However, it is debatable as to whether such integration is desirable given the current state-of-the-art of generalisation. In software engineering, much emphasis is placed on the coherence and coupling of software elements. At the current state of development of bendsimplification, it appears that there is an over-coupling of operations. Iterative bend elimination, like other similar geometric algorithms, can be enacted at the level of information generalisation and map modelling. Bend amalgamation and exaggeration appear to be aesthetic refinements which are more focused on display (image) generalisation. Close coupling of model and image generalisation limits the utility and applicability of the algorithm, unless the constituent processes can be knowledge-driven. The model of the visualisation process which underpins most modular dataflow visualisation systems, such as AVS, Iris Explorer and Khoros, deliberately uncouples the various stages in the visualisation process to facilitate end-user programming in visual thinking (Gallop, 1995). Although the pure dataflow model may not be the ideal systems architecture for Digital Cartography and GIS (Visvalingam, 1994), the separation of model- and image-based operations does offer greater flexibility.

At present, Bendsimplification, like the simpler algorithms, provides a single sequence of display-oriented solutions. Yet, it is well known that generalizations are purpose oriented and are constraint-driven. The same lines are conceived differently in scale-dependent and scale-independent generalizations. Uncoupling the constituent software elements into parameter-driven re-usable components offers opportunities for exploring multiple solutions.

Furthermore, the close coupling of a number of different operations within bendsimplification permits only a black-box approach to software testing. Researchers in computational geometry have pointed out that the implementation of even relatively simple geometric algorithms involves explicit consideration of the special cases (see Visvalingam and Whyatt, 1991). The black-box interface to the Bendsimplify option (as noted earlier) does not facilitate the performance of systematic hypothesis-driven tests. Consequently, a number of questions remain unanswered; for example:

1. ***To what extent are bends better indicators of features than triangles subtended by points?*** The intuitively appealing assumption, that bends are better, needs to be verified. Systematic evaluations require that all except one variable are kept constant. Since Visvalingam's iterative point elimination algorithm is driven by unweighted effective area, it would be revealing to test a similar process of iterative bend elimination using the same metric. This requires additional parameters to switch on/off

the shape weighting and other the other processes.

2. ***Is the universal application of the shape weighting justified?*** It would be equally useful to compare the outcome of running Visvalingam's algorithm with Wang and Muller's shape metric and weighting.

3. ***Is the shape weighting used in bendsimplification an appropriate conceptualisation of the impact of shape?*** There is a need to check this.

4. ***Is bend combination really necessary?*** Compared with Bendsimplify, Visvalingam's algorithm produces equivalent, if not better, typification of coastlines, without the necessity for programmed bend combination. Even the combination of just two bends involves too many ad-hoc decisions. What does bend combination offer over and above bend elimination? Again, it is not possible to evaluate this without the capacity to control the behaviour of the software through switches.

7. Conclusion

The Bensimplify option in Arc/Info was investigated to establish whether Bendsimplification offered a more incisive tool, than Visvalingam's algorithm, for research on line segmentation. The results revealed some major problems with the algorithm, and its implementation within Arc/Info 7.1.1, which reduce its utility and wider applicability. The results also show that despite using a very large number of points, contrary to Wang and Muller's intentions, the algorithm produces unbalanced and unacceptable results. Iterative bend elimination on its own will provide a different, and perhaps better, solution than Visvalingam's iterative point elimination algorithm. To test this, options are needed to switch constituent processes on/off; such options may also increase the utility, and not just the usability, of the software since the separation of model- and image-oriented processes would open up opportunities for generating multiple solutions. The various values driving the algorithm, such as the shape weighting, should also be parameterised. Some suggestions for improving the usability of the ***generalize*** command in Arc/Info were also noted (see Section 3). At the current state of development of bendsimplification, the ArcView systems architecture may provide a better environment, than Arc/Info, for its further research.

ACKNOWLEDGMENTS

All responsibility for the views expressed here rest with Mahes Visvalingam, who instigated this unfunded research. Simon Herbert investigated the Bendsimplify option in Arc/Info as a part of his 1997/98 Final Year Undergraduate Project in Computer Science. This paper was prompted by the results he obtained using large-scale road boundary data for roads (Herbert, 1998). We would like to thank John Farrow, formerly of Ordnance Survey for providing us with this data for research. Thanks are also due to Duncan Whyatt, a past PhD student within the CISRG, now a Lecturer in the Geography Department of Lancaster University, for technical advice on Arc/Info and for running the Carmarthen Bay coastline data through Arc/Info and alerting ESRI of the results. The data for Humberside was investigated as part of Simon Herbert's MSc research. We are grateful to Dan Lee and Zeshen Wang in ESRI for facilitating this study; they not only provided copies of papers on bendsimplification, but also debugged problems arising from the algorithm's implementation within Arc/Info 7.1.1. This feedback to others would not have been possible without their collaboration.

REFERENCES

Douglas, D. H., and Peucker, T. K. 1973. "Algorithms for the Reduction of the Number of Points Required to Represent a Digitised Line or its Caricature" *The Canadian Cartographer*, vol. 10, no. 2, pp. 112-122.

Gallop, J. 1994. "State of the Art in Visualization Software", In: *Visualization in Geographical Information Systems,* eds. Hearnshaw, H. M. and Unwin, D. J., John Wiley & Sons, pp. 42 - 47.

Herbert, S. 1998. "Extracting the Structure of Lines using Line Generalisation Algorithms" Unpublished Undergraduate Final Year Report in Computer Science, University of Hull, 1997/98.

Mandelbrot, B. 1983. *The Fractal Geometry of Nature,* W.H. Freeman, New York, 3rd edition.

McMaster, R. B. 1987. "Automated Line Generalisation" *Cartographica,* vol. 24, no. 2, pp. 74-111.

Visvalingarn, M., 1989, "Cartography, GIS and maps in perspective", *Cartographic J*, vol. 26, no. 1, pp 26 - 32.

Visvalingam, M., 1994. "Visualisation in GIS, Cartography and VISC", In: *Visualization in Geographical Information Systems,* eds. Hearnshaw, H. M. and Unwin, D. J., John Wiley & Sons, pp. 18 - 25.

Visvalingam, M. 1996. "Approximation, Generalisation and Deconstruction of Planar Curves", Cartographic Information Systems Research Group Discussion Paper 14, University of Hull, 12 pp.

Visvalingam, M. and Brown, C. I., 1999. "The deconstruction of teragons into decogons", *Computers & Graphics*, vol. 23, no. 1, in press.

Visvalingam, M., and Whyatt, J. D. 1991. "Cartographic Algorithms: Problems of Implementation and Evaluation and the Impact of Digitising Errors" *Computer Graphics Forum,* vol. 10, no. 3, pp. 225-235.

Visvalingam, M., and Whyatt, J. D. 1993. "Line Generalisation by Repeated Elimination of Points" *Cartographic J,* vol. 30, no. 1, pp. 46-51.

Visvalingam, M. and Williamson, P. J. 1995. "Simplification and Generalisation of Large Scale Data for Roads : a comparison of two filtering algorithms", *Cartography and Geographical Information Systems* vol. 22, no 4, 264 - 275.

Wang, Z. 1996. Manual versus Automated Line Generalization, In: *Proceedings of GIS/LIS '96,* Denver, Colorado, 94 - 106.

Wang, Z. and Muller, J-C. 1998. "Line Generalization based on Analysis of Shape Characteristics". *Cartography and Geographical Information Systems,* 25 (1): 3 - 15.

Whyatt, J. D. 1991. "Visualisation and Re-evaluation of Line Simplification Algorithms" Unpublished Ph D Thesis, University of Hull, 259 pp.

Whyatt, J. D. and Wade, P. 1988. "The Douglas-Peucker Line Simplification Algorithm" *Bulletin of the Society of University Cartographers* vol. 22, no. 1, 17 - 25.