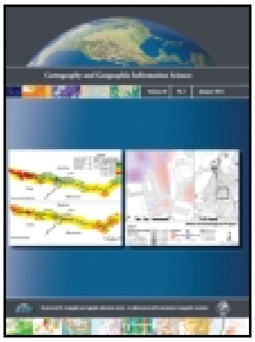


Cartography and Geographic Information Science

ISSN: 1523-0406 (Print) 1545-0465 (Online) Journal homepage: http://www.tandfonline.com/loi/tcag20

A Computer Science Perspective on the Bendsimplification Algorithm

Mahes Visvalingam & Simon Herbert

To cite this article: Mahes Visvalingam & Simon Herbert (1999) A Computer Science Perspective on the Bendsimplification Algorithm, Cartography and Geographic Information Science, 26:4, 253-270, DOI: 10.1559/152304099782294177

To link to this article: http://dx.doi.org/10.1559/152304099782294177

A Computer Science Perspective on the Bendsimplification Algorithm

Mahes Visvalingam and Simon Herbert

ABSTRACT: The primary aim of this study was to evaluate whether the use of bends provides a better basis than point elimination for research on line structuring. These investigations were undertaken using Arc/Info 7.1.1. Comparative 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 therefore provides a brief review of these three algorithms. A more detailed conceptual and empirical evaluation of the bendsimplification system follows, highlighting some problems with implementing the system in Arc/Info. The paper then questions the value of over-coupling model- and image-oriented generalization processes within the black-box bendsimplification 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 provides some pointers for further research. With respect to the main aim of the research, the evidence suggests that bendsimplification is less useful for line segmentation than Visvalingam's algorithm. Further research is needed to assess the value of the iterative bend elimination operator within bendsimplification.

KEYWORDS: Line generalization, bendsimplification, Arc/Info, Visvalingam's algorithm

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 generalization of lines. Visvalingam's iterative removal of triangular features, subtended by points and their neighbors, results in the progressive elimination of scale-related features (Whyatt 1991; Visvalingam and Whyatt 1993). The study reported here was prompted by the idea that the iterative elimination of convex and concave bends (Wang 1996) might be more useful for revealing line structure than iterative point elimination. Bends are defined as sections of curves between two inflection points. The idea of iterative bend elimination has been incorporated within the bendsimplification approach (Wang and Muller 1998), which was available as an option within Arc/Info 7.1.1. Wang and Muller's (1998) bendsimplification system detects bends in lines to drive the line generalization process; it was proposed as a form of structure-based generalization. Herbert (1998) explored the utility of the Bendsimplify option in a project concerned with line segmentation. The results were unexpected and prompted further evaluation of this new algorithm by the authors.

Mahes Visvalingam is Senior Lecturer in Computer Science, Cartographic Information Systems Research Group, Department of Computer Science, University of Hull, Hull, HU6 7RX United Kingdom. Tel: 01482 465295; Fax: 01482 466666. E-mail: <M.Visvalingam@dcs.hull.ac.uk>. **Simon Herbert** was M.Sc. research student in Computer Science in 1998/99.

In this paper, "bendsimplification" refers 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.

Comparative experimental results suggest that the Bendsimplify algorithm may not be as widely applicable as the much simpler geometric filters, such as the Douglas-Peucker or Visvalingam algorithms. The paper therefore provides a brief review of these three algorithms and provides a more detailed conceptual and empirical evaluation of the bendsimplification algorithm. Some problems with the implementation of the algorithm in Arc/Info are also identified.

The paper questions the value of over-coupling model- and image-oriented generalization processes within the black-box bendsimplification 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 provides some pointers for further research. With respect to the main aim of the research, the evidence suggests that bendsimplification is less useful for line segmentation than Visvalingam's algorithm. Because iterative bend elimination is embedded as one component within a rule-based system, this study could not conclusively establish whether it is more useful for line structuring than Visvalingam's algorithm.

Review of Line Generalization Algorithms

Although the main focus of the study was an evaluation of the bendsimplification system for research into line structuring, comparative results for the Douglas and Peucker (1973) and Visvalingam's (Visvalingam and Whyatt 1993) algorithms are provided here for the benefit of readers. A brief review of the three algorithms is provided, starting with the oldest in the set, before evaluating the Arc/Info Bendsimplify option and its output.

The Douglas and Peucker (1973) algorithm was initially developed for weeding out superfluous points in lines. The hierarchical version of the Douglas-Peucker algorithm (used here) was independently proposed for the same purpose by Ramer (1972), namely for the approximation of a curve by a minimum number of points. In essence, the method consists of recursively splitting the line at the point furthest from the current base line, which connects the first and last points of the subline being examined. Normally, this process is recursed until the maximum offset is less than some pre-defined tolerance.

In Wade's implementation of this algorithm (reported in Whyatt and Wade 1988), the offset distance which led to the selection of all points may be recorded, by using a zero tolerance, so that the line may be more flexibly filtered at run time. Visvalingam and Whyatt (1991) noted the special cases which arise and how they are dealt with in Wade's implementation. The Douglas-Peucker algorithm was considered by Peucker (1975) and used by Ballard (1981) for constructing band and strip trees, respectively, for representing lines to facilitate geometric operations such as overlay analysis. Buttenfield (1986), in turn, based her analysis of line signatures on strip trees. Despite nearly 30 years of research, this method has not led to a satisfactory scheme for structuring lines for line generalization and is included here mainly for the sake of completeness.

Visvalingam's (Visvalingam and Whyatt 1993) algorithm was conceptualized as an algorithm for eliminating features. Given that we do not have the know-how for segmenting meaningful in-line features, the algorithm focused on the elimination of triangular features as a first step. As it stands, it is also a point filtering algorithm like the Douglas and Peucker (1973) algorithm, However, Visvalingam's algorithm was designed for caricatural generalization, not for minimal simplification.

While the Douglas-Peucker algorithm operates by selecting points, Visvalingam's algorithm focuses on iterative point elimination. On each iteration, the point which contributes the least information (however defined) is eliminated. This algorithm may be driven by any metric, but research to-date suggests that the effective area of a point (which is the area subtended by the point and its two neighbors) provides the best measure of significance for 2D lines; it was also found to be the best measure for producing artistic sketches of GRID Digital Elevation Models (Visvalingam and Dowson 1998). Tests to-date suggest that the method is capable of typification and caricatural generalization since it progressively eliminates features within features in their entirety while preserving the shape of the retained features. Since point significance is assessed relative to the part-processed rather than the original line, sections of the line become detached. In general, these detached sections correspond to semantic entities, suggesting that the algorithm may be used for line structuring.

As with the Douglas-Peucker method, Visvalingam's algorithm makes it possible to tag each point with the effective area which led to its elimination so that the line may be filtered at run time. This may also be used to structure the line into a tree of points. However, line structuring for generalization, as opposed to spatial search, should be concerned with segmenting and assembling in-line features into a structure which has semantic meanings. Visvalingam and Whyatt (1993) and Visvalingam and Williamson (1995) noted that this method provides clues for such meaningful segmentation of lines. Herbert (1998) attempted to extract the in-line components. He investigated the Bendsimplify algorithm because it operated on bends rather than points, but found that the results from Visvalingam's algorithm were more promising.

Wang's (1996) experimental system for line generalization was based on four guidelines for manual generalization as provided by the Swiss Society of Cartography in 1977 (see Wang and Muller 1998).

These "rules" include the retention of isolated small bends, the combination of similar adjacent bends, and the exaggeration of retained bends. Wang and Muller:

- Initially equated the recognition of structure with the recognition of bends. "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).
- Used bend characteristics (e.g, shape-weighted size of bends, isolated occurrence of bends, and proximity of bends) to trigger generalization operators such as bend elimination, bend combination, and bend exaggeration. They stated that "Most existing line generalization solutions are based on geometric processing without previous shape analysis..." (p 5); and, that "The method we propose is based on shape analysis; it uses information on the shape of the bends along a line and their relationships to select appropriate generalization operations. Brassel and Weibel (1988) called the shape analysis a 'structure recognition...'" (p. 4).
- Stated 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. Some preliminary results, obtained using the Arc/Info Bendsimplify option, were sent to Wang and Lee (personal communication, February 1998) who clarified that the concept of bends referred only to simple bends and that it did not include deep bays with branches.
- Concluded with "There is also a need for additional cartographic rules for line structure recognition to enable more sophisticated operations" (p. 14).

We were not entirely clear as to precisely what Wang and Muller meant by structure recognition. Their paper starts off by suggesting that the major task is one of bend identification to reveal line structure. Its conclusion implies that additional supplementary rules are needed. So, one aim of our study was to derive a better understanding of their work.

The above three algorithms are compared using three sets of data, namely large-scale data for roads, small-scale data for coastlines and the Koch curve. Fractal curves have proved to be good test lines for evaluating the geometric properties of line generalization 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.

Given the aims of bendsimplification, the mathematical measures of evaluation (proposed by McMaster (1987) for measuring the goodness of fit of approximations) are inappropriate here. Therefore, this evaluation is based largely on visual judgment. Although the usual process of comparing output containing an equivalent number of points is not strictly applicable, given the aims of bendsimplification, it is nevertheless helpful and is used here.

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). Because 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. As 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 problems, as noted below.
2. *There is also a lack of user access to the metric information used by the software to drive the generalization* 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. 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.

Observations

Large-scale Data for Roads

Figure 1 shows sample output from the three algorithms for the 1:1250 large-scale data for a road boundary. This data set, consisting of 270 points, was previously used by Visvalingam and Williamson (1995) to compare the Douglas-Peucker with the Visvalingam point filtering algorithm. They concluded that the Douglas-Peucker algorithm was better for minimal simplification (see inset c in Figures 1i and 1ii) but that Visvalingam's algorithm was better for generalization (insets d-f in Figures 1i and 1ii).

The retention of extreme points by the Douglas-Peucker method leads to the retention of all the roads and a distortion of their shapes in Figure 1i. In contrast, Visvalingam's algorithm drops smaller features in their entirety and retains the shape of the major features. The results from the Bendsimplify option (Figure 1iii) speak for themselves: Even with 169 points, the shape of the minor road is grossly distorted, whereas the Douglas-Peucker and Visvalingam's algorithms give excellent approximations at this level.

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 with shape distortion and with many more points. Visvalingam's algorithm produces much better results; compare the number of points in Figure 1iiic with Figure 1iie and in Figure 1iiie with 1iif. Visvalingam's algorithm defines the broad shape of the main road boundary before bringing in the branch road; Bendsimplify eliminates the former (Figure 1iiid) before the latter (Figure 1iiie). Bendsimplify's exaggeration of branch roads is not appropriate here (Figures 1iiia to iiid); 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.

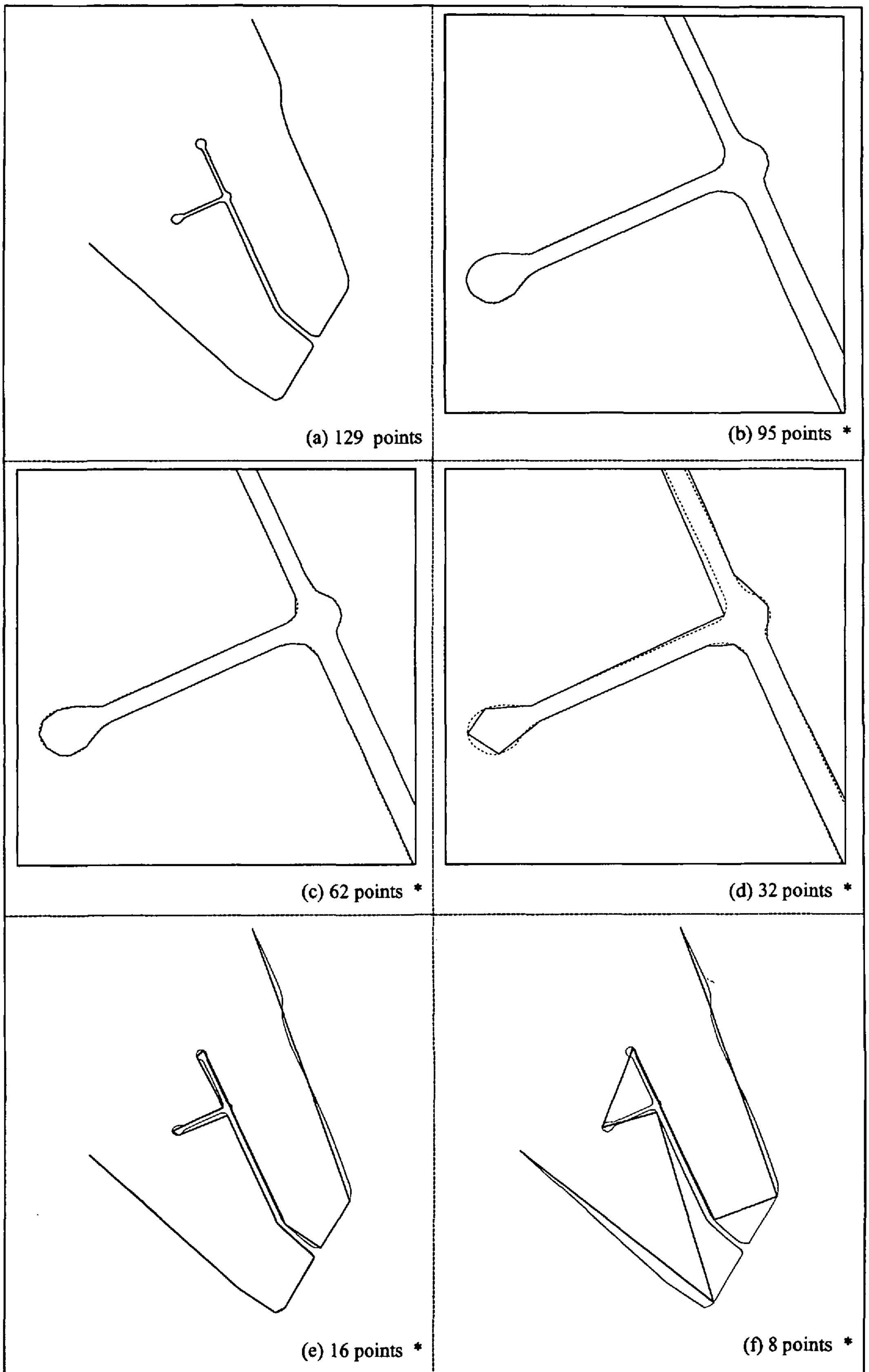
Coastlines

Arc/Info's Bendsimplify option was tested using the coastline of Carmarthen Bay, previously used by Visvalingam and Whyatt (1993). Whyatt (personal communication, 1998) provided some Bendsimplify results based on these test data. The results did not compare well with those produced by Visvalingam's algorithm, and those with 25 percent or less points were particularly 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 generalization 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 percent), and even just four 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). This coastline does not have a complex hierarchy of features within features as does the Carmarthen Bay, but it has two meandering rivers (detail in Figure 3). The Douglas-Peucker and Visvalingam algorithms can filter down to even 2.5 percent of points (Figure 2), although the results of the former are adversely affected by the extreme points on the River Ouse (Figure 3i).

Visvalingam's algorithm achieves good minimal simplification of the head of the estuary using less than 25 percent of points (Figure 3iia); with 5 and 2.5 percent of points it is possible to achieve the style of depiction provided in atlas maps (see Figures 3iic and d). The mouth of the rivers may be enlarged in **visual** mapping; but this introduces unnecessary complications for line segmentation, which is a part of **digital** mapping (Visvalingam 1989; 1999). The program of research within the CISRG[1] assumes that database-oriented digital mapping could facilitate image-oriented interactive visual mapping.

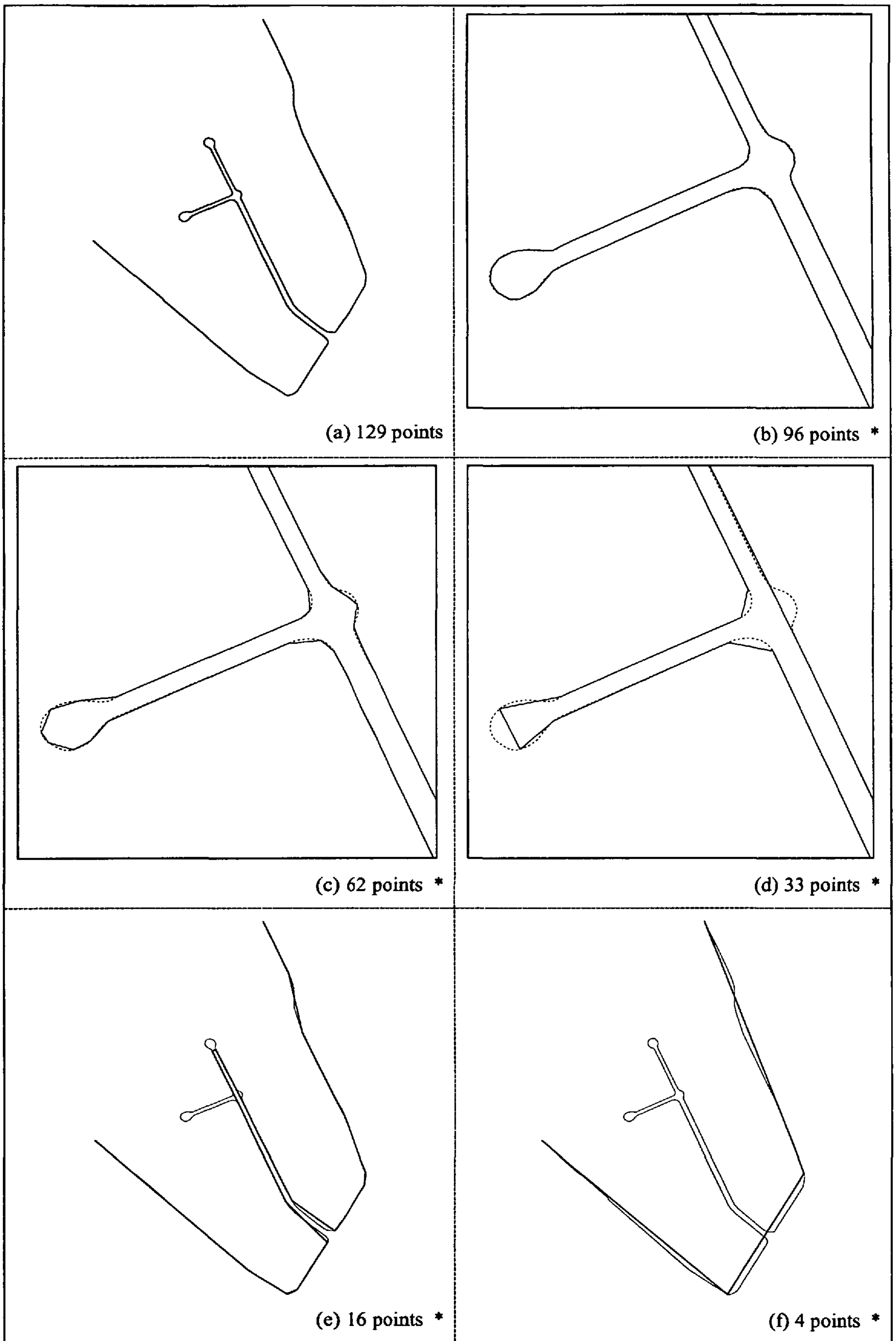
[1] CISRG= Cartographic Information Systems Research Group.



i) Douglas-Peucker algorithm (* Selected figures from Visvalingam and Williamson, 1995)

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

Figure 1 Continued / . . .



ii) Visvalingam's algorithm (* Selected figures from Visvalingam and Williamson, 1995)

Figure 1 Continued / . . .

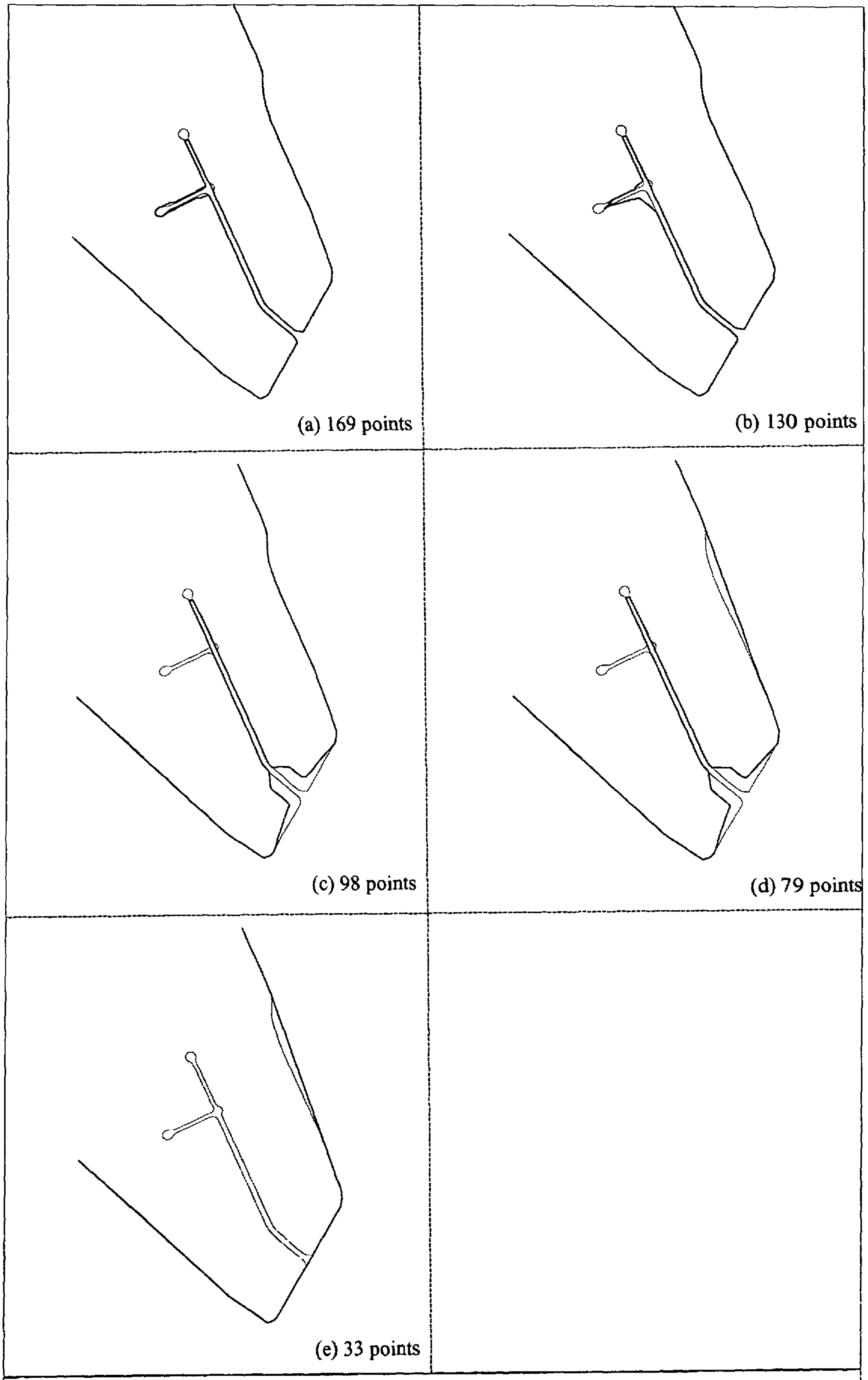


Figure 1 iii) Bendsimplify option in Arc/Info 7.1.1

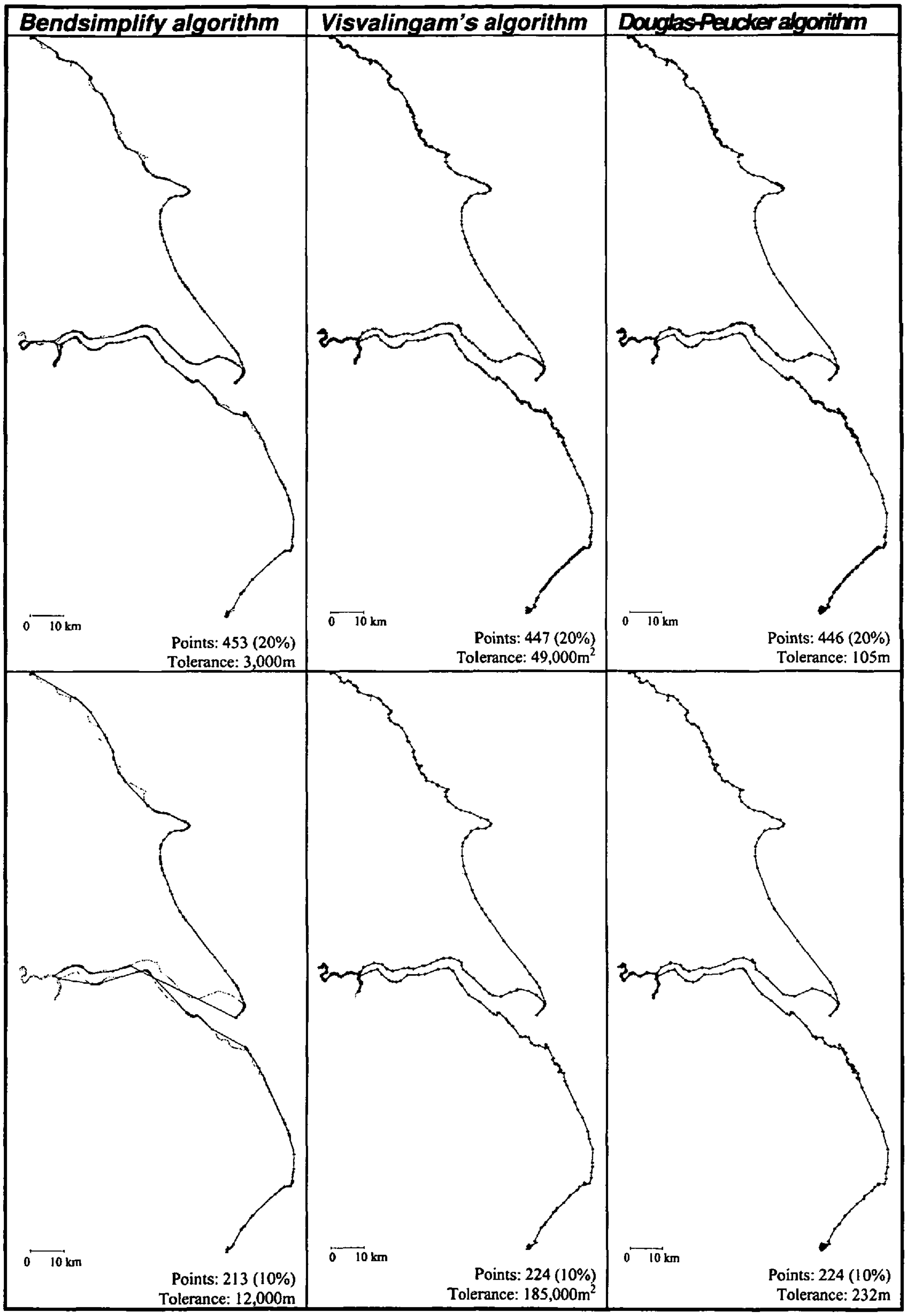


Figure 2 Continued / . . .

The bendsimplification algorithm produces very unbalanced results (Figure 2). Because Arc's segmentation of the line into 500-point sections may have affected the simplification process, the 500-point section of the line at the head of the Humber Estuary was studied to remove Arc intervention (Figure 3iii).

By its very name, bendsimplification is said to be designed for minimal simplification—not caricatural generalization. Hence, the comparisons

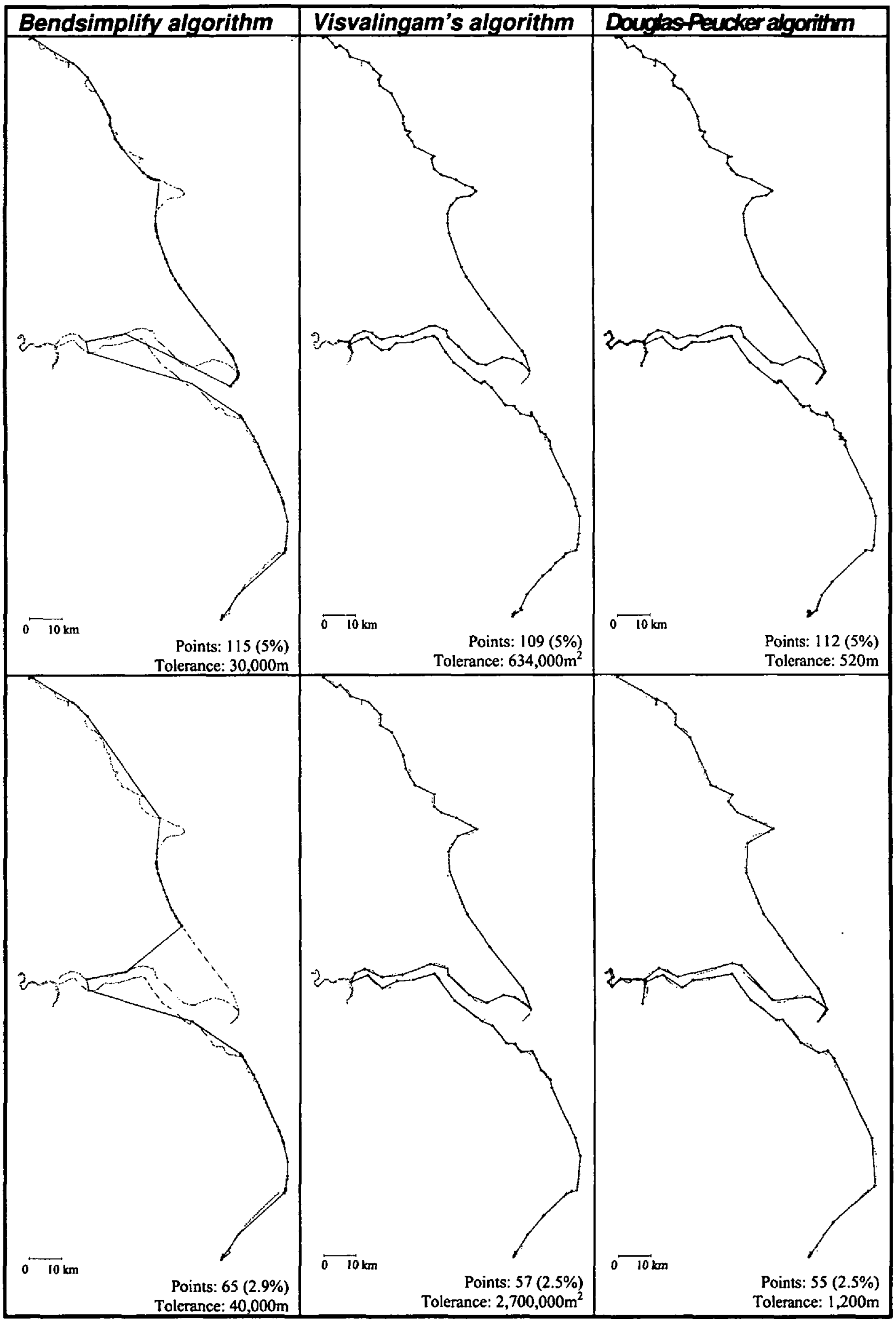


Figure 2. Generalizations of the 1:50000 coastline of Humberside, U.K. Data source: Ordnance Survey, Crown Copyright reserved.

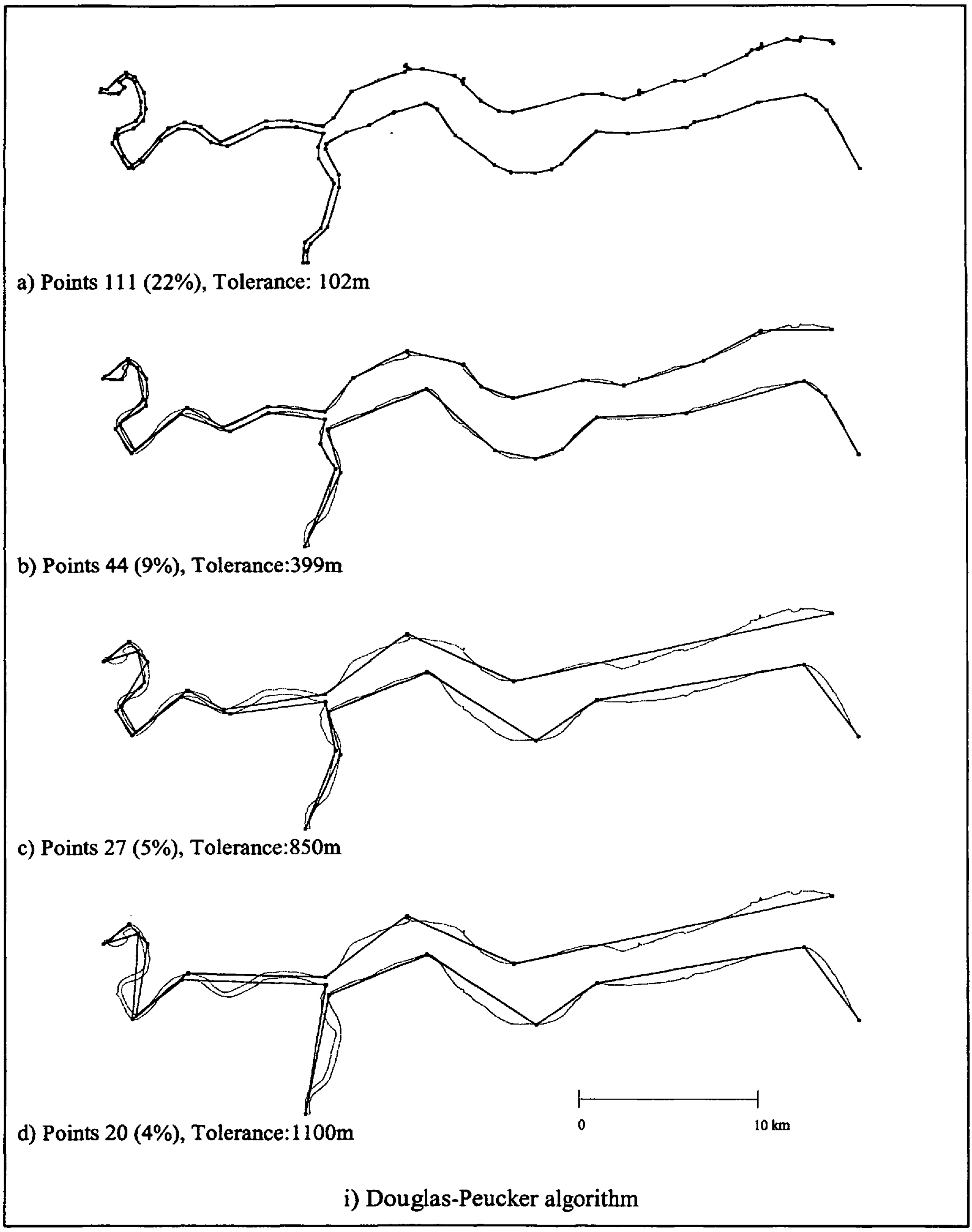


Figure 3 Continued / . . .

made in Figure 2 may appear irrelevant to some. However, Figure 3iii shows that even if we reduce the tolerance to retain over 50 percent of points, the output is still unbalanced. Figure 3iiia, with 58 percent of points, shows the sort of minimal simplification provided by Visvalingam's algorithm with just 22 percent of points (Figure 3iia). Figure 3iiib, with 57 percent 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 3iiic (with 35 percent of the points) where it results in a lake attached to the head of the

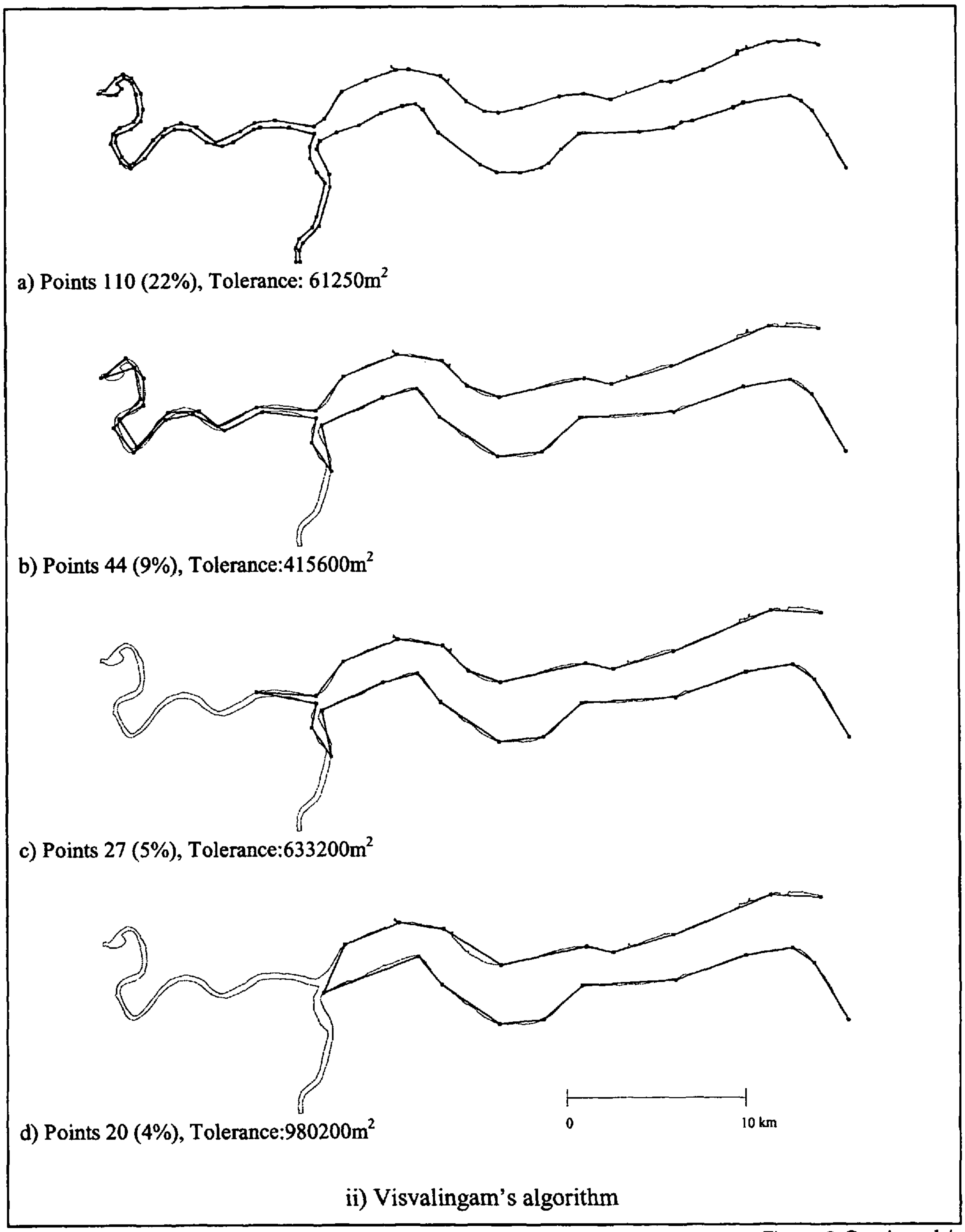


Figure 3 Continued / . . .

estuary by two straight lines. This is contrary to a guideline provided by the Swiss Society of Cartography that, *Non-straight lines should not be replaced by geometrically straight lines* (Wang and Muller 1998, p. 4). Thus, the unbalanced results observed at gross levels of filtering are also present at the 35 percent level of filtering which generally supports excellent minimal simplification with, especially, the Douglas-Peucker algorithm (Figure 3i).

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 3iiic, as the roads had been in Figure 1iii. Visvalingam's algorithm with 5 percent of points 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 percent of the Humberside points in the case of Visvalingam's algorithm (Figure 3iib), Figure 3iiib (with over 50 percent 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 clues for selecting scale-related styles of generalization 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 generalization.

The Quadric Koch Island

The Bendsimplify option was tested using fractals. The rectangular Koch island was chosen because it makes it easier to investigate the reasons for Bendsimplify's retention of a large number of points. Fractal curves are generated by repeatedly

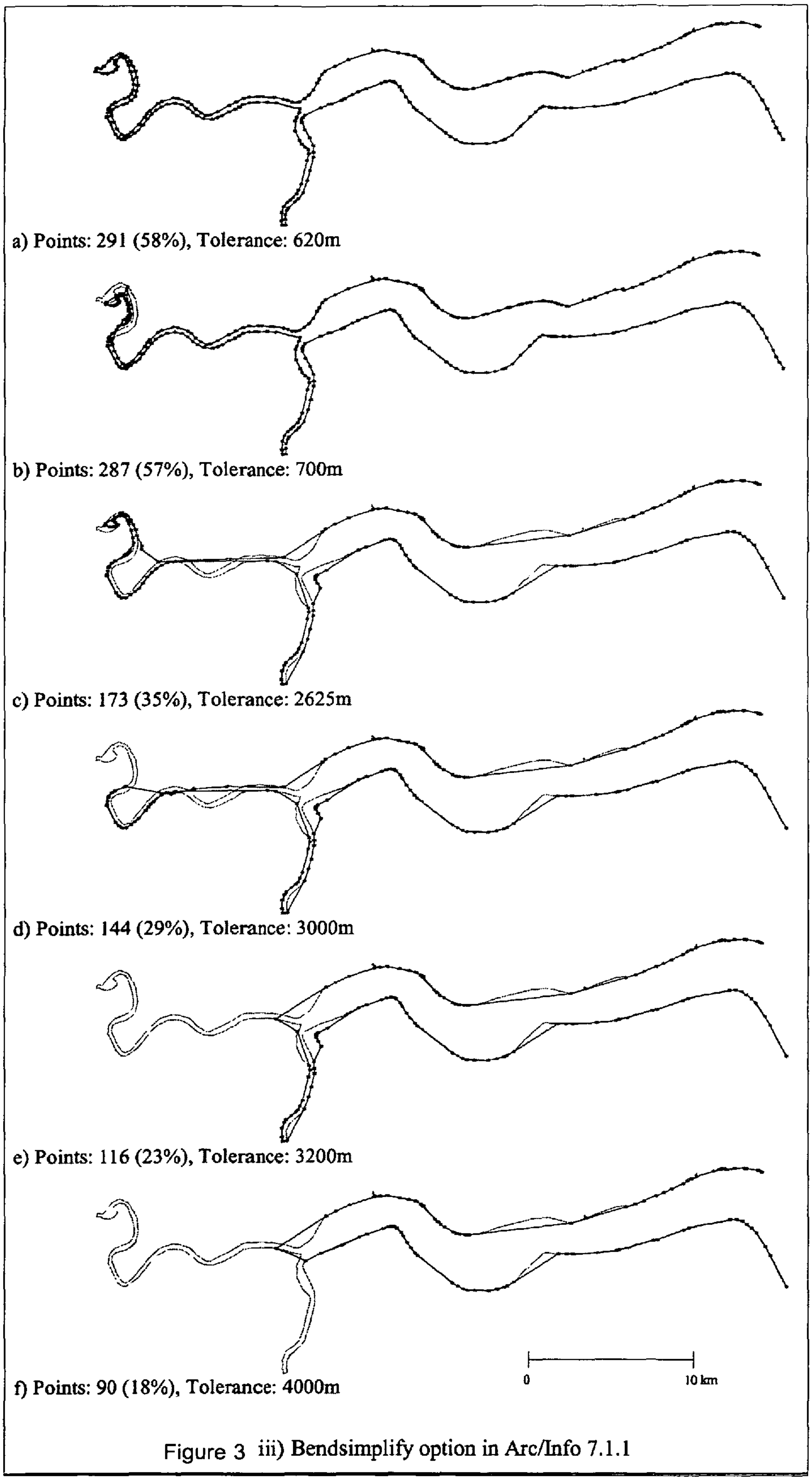


Figure 3. Generalizations of the Humber Estuary—enlarged. Data source: Ordnance Survey, Crown Copyright reserved.

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 generalization 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 generalization, respectively. Whereas manual generalization 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 were used as test lines in this study. The level 1 teragon was initially considered because it only has simple "bends" with no hierarchy of nested bends. Visvalingam and Brown (1999) showed that a variety of decogons may be derived 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 alone, which is identical to that produced by Visvalingam's algorithm with effective area. However, Bendsimplify 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 just 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).

Figures 4-6 show the results obtained from the level-2 teragon (consisting of just 256 points) using the Douglas-Peucker and Visvalingam's algorithms and the Bendsimplify option, respectively. A detailed analysis of Figures 4 and 5 are provided elsewhere (Visvalingam and Brown 1999); here it is sufficient to note that the 4-fold symmetry was maintained throughout progressive deconstruction. The Douglas-Peucker algorithm is able to recover a rotated and distorted version of the generator (Figure 4).

Figure 6a shows the first reduction of the line by Bendsimplify. Whereas Visvalingam's algorithm only uncovers the level-1 equivalent 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 4-fold symmetry. So, it is likely that the loss of symmetry arises from bend amalgamation as 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) and Visvalingam and Brown (1999) 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. Digitizing 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 (Figure 4). Although the latter has a propensity to create spikes where none exist and produce an unbalanced generalization 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 Figures 4 and 5 only consist of 8 and 12 points, respectively. Bendsimplify does not cull redundant points.

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

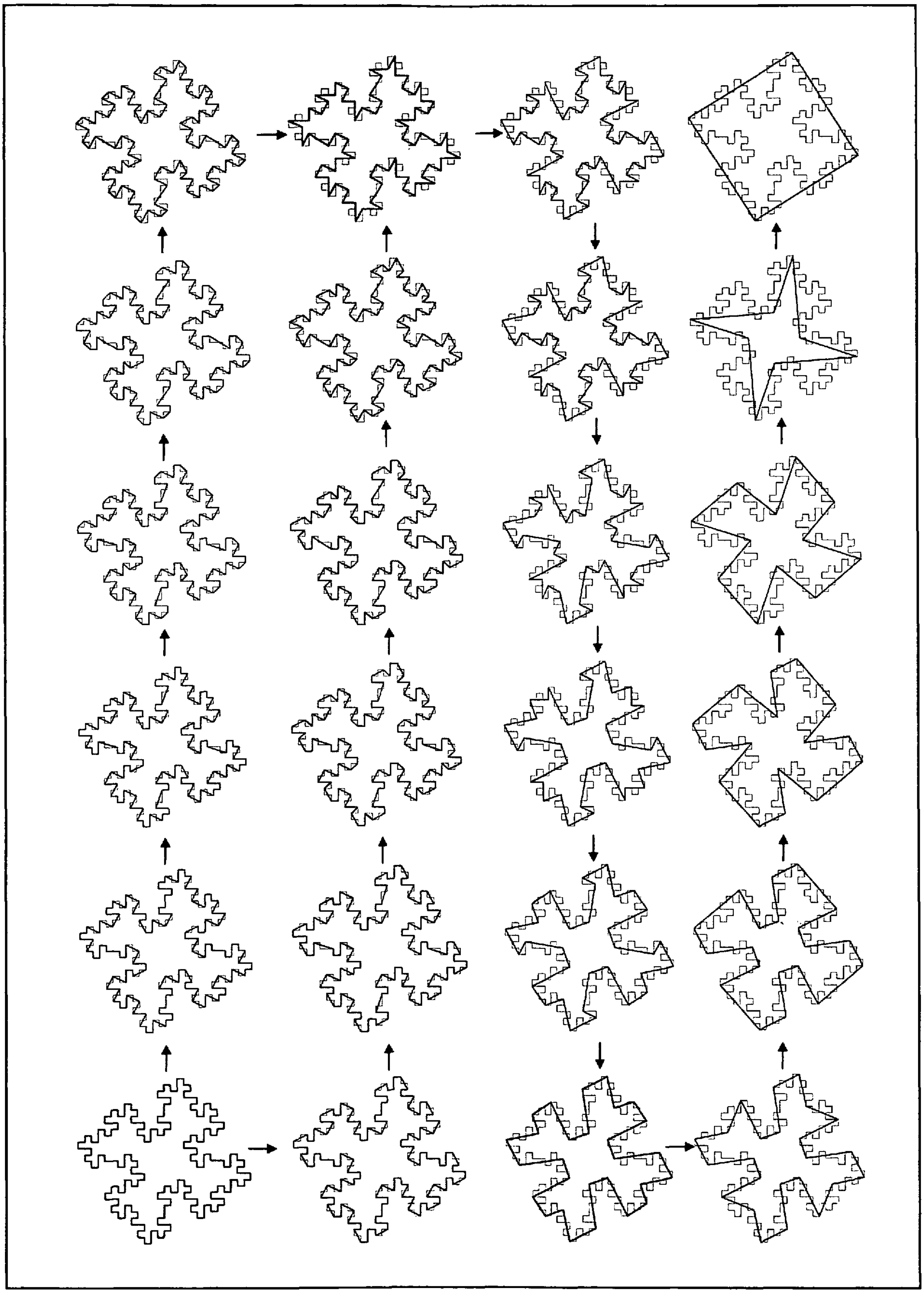


Figure 4. Deconstruction of the level 2 quadric Koch teragon by the Douglas-Peucker algorithm. [Reprinted from Visvalingam and Brown (1999), with permission from Elsevier Science.]

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 that "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 claimed to be distinctive for 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 generalization. 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 generalization and map modeling (Visvalingam 1999). Bend amalgamation and exaggeration appear to be aesthetic refinements which are more focused on display (image) generalization.

Close coupling of model and image generalization limits the utility and applicability of the bendsimplification algorithm, unless the constituent processes can be knowledge driven. The model of the visualization process which underpins most modular data flow visualization systems, such as AVS, Iris Explorer and Khoros, deliberately uncouples the various stages in the visualization process to facilitate end-user programming in visual thinking (Gallop 1994). Although the pure data flow 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, as suggested by Weibel (1995).

At present, bendsimplification, like the simpler filtering algorithms, provides a single sequence of display-oriented solutions. Yet, it is well known that generalizations are purpose oriented and 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, such as:

- *What precisely does line structuring mean within bendsimplification?*

  This is not entirely clear. As noted earlier, Wang and Muller refer to in-line structure recognition in various ways, namely as "detecting bends," as "shape analysis," or as "cartographic rules." Their flow chart shows that any one of bend operators —exaggeration, combination, and elimination—may be triggered (in that order) by the shape of bends and their relationship to neighboring bends.

  Bends are clearly at the core of bendsimplification, but, their system appears to re-shape the line continually on aesthetic and not just structural criteria. The aesthetic judgments may be invoked at any step in the generalization process and could prejudice the subsequent judgment of structure and further re-shaping of the line. Furthermore, the output of this iterative process is sensitive to a single tolerance (which has to be found by trial and error). Consequently, bendsimplification appears to focus on the ephemeral structure of a mutating line and does not directly address the structure of the input line. If this is so, then bendsimplification falls within the category of display generalization. The output of the Bendsimplify option also indicates that bendsimplification is not very useful for revealing the in-line structure of the input lines.

- *Is iterative bend elimination better than point elimination?*

  Our interest in bendsimplification was stimulated by Wang's (1996) earlier paper on iterative bend elimination. Hoffman and Richards (1984), working in the field of pattern recognition, segmented contours into convex (and concave) parts (which correspond loosely to bends) at extremes of concavities to resolve ambiguous reversing figures, such as the Rubin vase/face. Given that bends have psychological significance, Wang's attempt to eliminate bends instead of points is

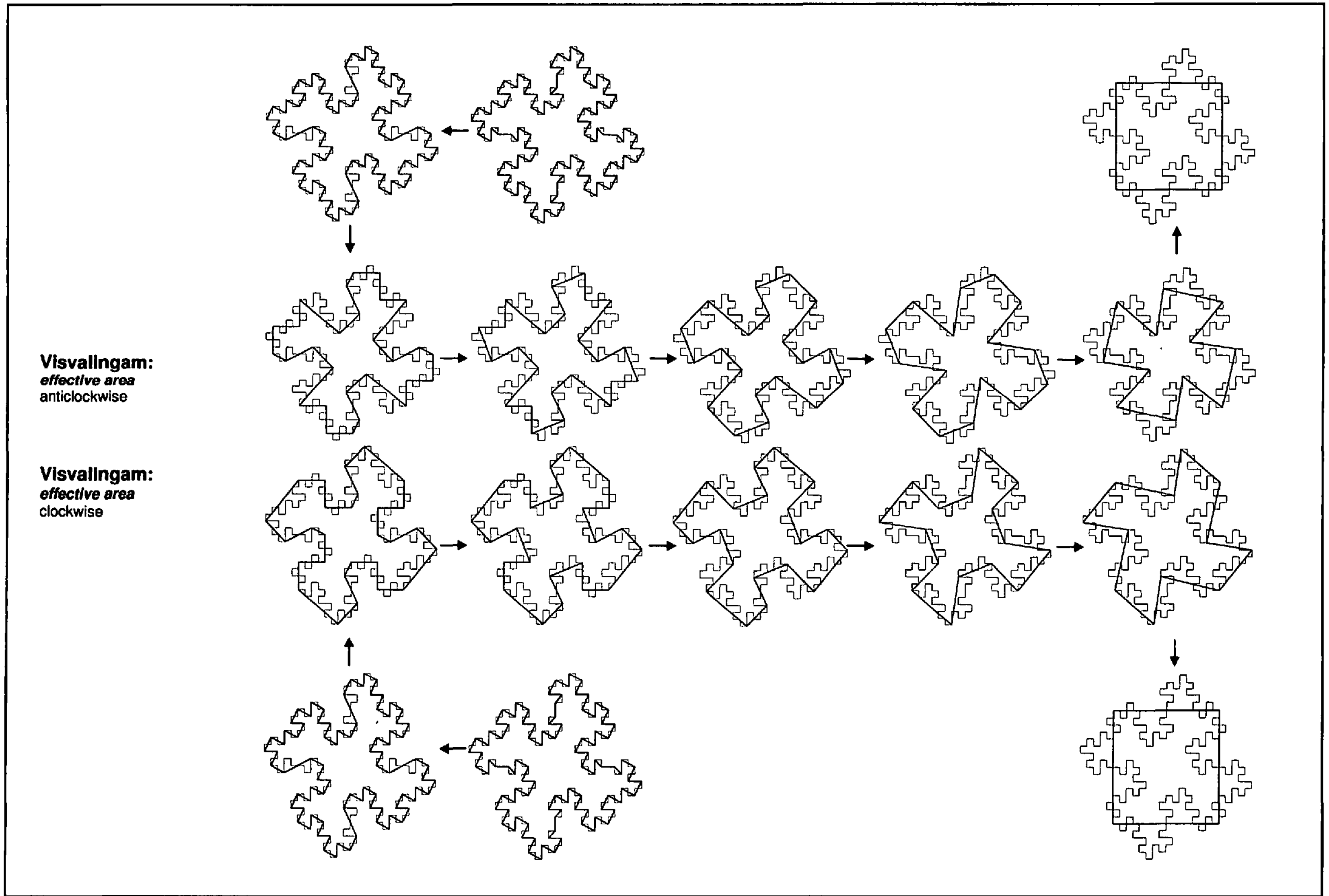


Figure 5. Deconstruction of the level 2 quadric Koch teragon by Visvalingam's algorithm using effective area. [Reprinted from Visvalingam and Brown (1999), with permission from Elsevier Science.]

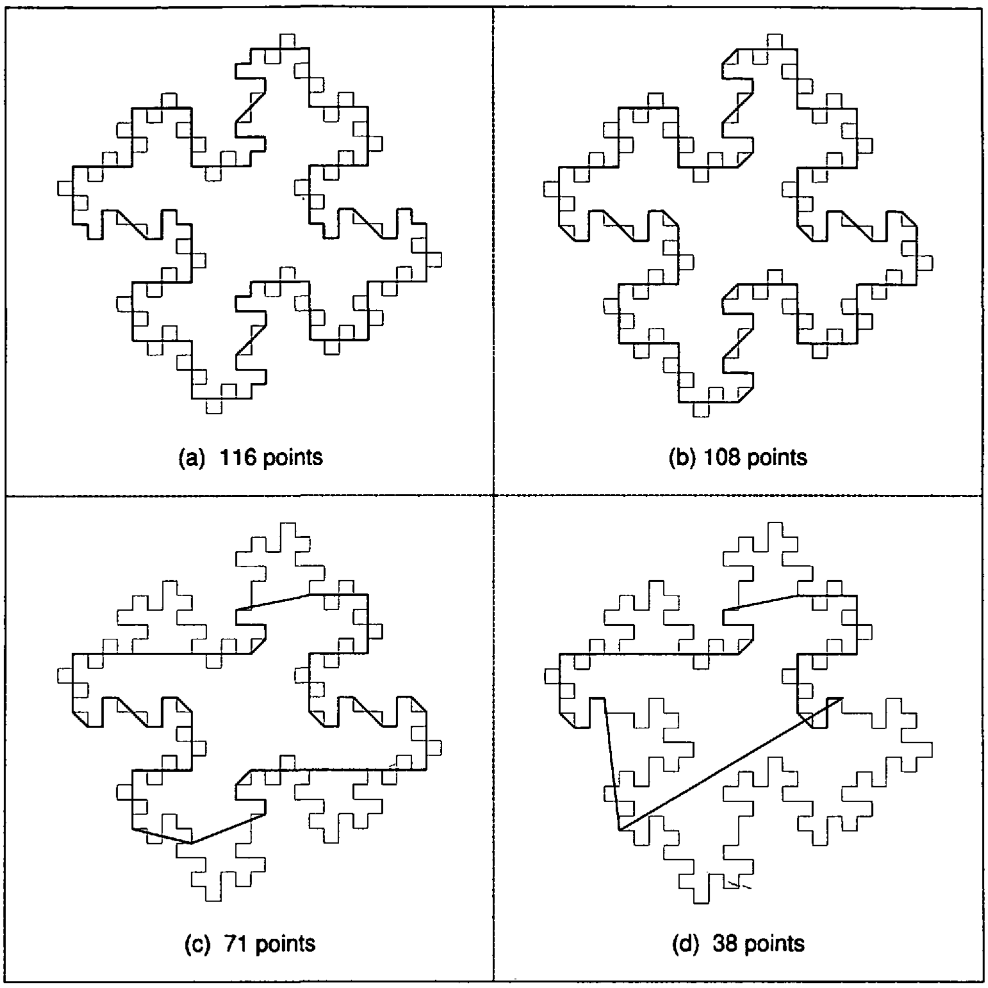


Figure 6. Bendsimplification of the level 2 quadric Koch teragon.

interesting and merits independent, rigorous verification with demanding test data. The convoluted nature of many coastlines (such as the Carmarthen Bay) and the errors and inadequacies in digital data capture and in their processing (e.g., prior weeding), can distort results. Unfortunately, because of the black-box nature of the Bendsimplify option, our results cannot be regarded as conclusive.

- *Is the universal application of shape weighting justified?*
  It would be useful to compare the outcome of running Visvalingam's algorithm with Wang and Muller's shape metric and weighting. Equally, since Visvalingam's iterative point elimination algorithm gives good results with unweighted effective area, iterative bend elimination should be tested using the same metric.
- *Is the shape weighting used in bendsimplification an appropriate conceptualization of the impact of shape?* There is a need to check this.
- *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 behavior of the software through switches.

Conclusion

The Bendsimplify option in Arc/Info was investigated to establish whether bendsimplification offers a more incisive tool than does 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 reduces its utility and wider applicability. The results also show that despite retaining 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 may provide a different, and perhaps better, solution than does 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 parameterized. Some suggestions for improving the usability of the **generalize** command in Arc/Info were also noted. At the current state of development of bendsimplification, its provision within the ArcView systems architecture may provide a better environment for further investigations than does Arc/Info.

ACKNOWLEDGMENTS

Simon Herbert investigated the Bendsimplify option in Arc/Info as part of his 1997/98 Final Year Undergraduate Project in Computer Science, under the supervision of Mahes Visvalingam. This paper was prompted by the results he obtained using large-scale road boundary data (Herbert 1998). We would like to thank John Farrow, formerly of Ordnance Survey, for giving us the data for research. The Humberside coastline data were investigated as part of Simon Herbert's MSc research.

Thanks are also due to past research students within the Cartographic Information Systems Research Group. We are grateful to Phil Wade for permission to use his implementation of the Douglas-Peucker algorithm and to Chris Brown for his assistance in the deconstruction of fractals. Dr. Duncan Whyatt, now a Lecturer in the Geography Department of Lancaster University, provided technical advice on Arc/Info. He analyzed the Carmarthen Bay coastline data with Arc/Info and alerted ESRI of the results. We are grateful to Dan Lee and Zeshen Wang in ESRI for facilitating this study; they not only provided pre-publication 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

Ballard, D. H. 1981. Strip trees : A hierarchical representation for curves. *Communications of Assoc. for Comp. Machinery* 24(5): 310-21.

Buttenfield, B. 1986. Digital definitions of scale-dependent structures. *Proc. Auto Carto London*, U.K. Vol.1, pp. 497-506.

Brassel, K., and R. Weibel. 1988. A review and framework of automated map generalisation. *International Journal of Geographical Information Systems* 2(3): 229-44.

Douglas, D. H., and T. K. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. *The Canadian Cartographer* 10(2): 112-22.

Gallop, J. 1994. State of the art in visualization software. In: Hearnshaw, H. M., and D. J. Unwin (eds), *Visualization in geographical information systems*. Chichester, U.K.: John Wiley & Sons. pp. 42-7.

Herbert, S. 1998. Extracting the structure of lines using line generalization algorithms. Unpublished Undergraduate Final Year Report in Computer Science, University of Hull, 1997/98.

Hoffman, D. D., and W. A. Richards. 1984. Parts of recognition. *Cognition* 8: 65-96.

Mandelbrot, B. 1983. *The fractal geometry of nature* (3rd ed.). New York, New York: W.H. Freeman.

McMaster, R. B. 1987. Automated line generalization. *Cartographica* 24(2): 74-111.

Peucker, T. K. 1975. A theory of the cartographic line. In: *Proc. Auto Carto 2*, pp. 508-18.

Ramer, U. 1972. An iterative procedure for polygonal approximation of planar closed curves. *Computer Graphics and Image Processing* 1: 244-56

Visvalingam, M. 1989. Cartography, GIS and maps in perspective. *Cartographic Journal* 26(1): 26-32.

Visvalingam, M. 1994. Visualisation in GIS, Cartography and VISC. In: Hearnshaw, H. M., and D. J. Unwin (eds), *Visualization in geographical information systems*. Chichester, U.K.: John Wiley & Sons. pp. 18-25.

Visvalingam, M. 1996. Approximation, generalization and deconstruction of planar curves. Cartographic Information Systems Research Group Discussion Paper 14, University of Hull, 12 pp.

Visvalingam, M. 1999. Aspects of line generalisation. Position paper for Third Workshop on Progress in Automated Map Generalization, ICA Working Group on Map Generalization, Ottawa, Canada, 12-14 August, 1999. [http://www2.dcs.hull.ac.uk/CISRG/ica/ica.htm].

Visvalingam, M., and C. I. Brown. 1999. The deconstruction of teragons into decogons. *Computers & Graphics* 23(1):155-67.

Visvalingam. M., and K.Dowson. 1998. Algorithms for sketching surfaces. *Computers & Graphics* 22(2-3): 269-280.

Visvalingam, M., and J.D. Whyatt, J. D. 1991. Cartographic algorithms: Problems of implementation and evaluation and the impact of digitising errors. *Computer Graphics Forum* 10(3): 225-35.

Visvalingam, M., and J. D. Whyatt. 1993. "Line generalization by repeated elimination of points. *Cartographic Journal* 30(1): 46-51.

Visvalingam, M., and P.J. Williamson. 1995. Simplification and generalization of large scale data for roads : A comparison of two filtering algorithms. *Cartography and Geographic Information Systems* 22(4): 264-75.

Wang, Z. 1996. Manual versus automated line generalization, In: *Proceedings of GIS/LIS '96*, Denver, Colorado. pp. 94-106.

Wang, Z., and J-C. Muller. 1998. Line generalization based on analysis of shape characterstics. *Cartography and Geographic Information Systems* 25(1): 3-15.

Weibel, R. 1995. Three essential building blocks for automated generalization. In: Muller J-C., J-P. Lagrange, and R. Weibel (eds), *GIS and generalization: Methodology and practice*. London, U.K.: Taylor & Francis. pp. 56-69.

Whyatt, J. D. 1991. Visualization and re-evaluation of line simplification algorithms. Unpublished PhD thesis, University of Hull. 259 pp.

Whyatt, J. D., and P. Wade. 1988. The Douglas-Peucker line simplification algorithm. *Bulletin of the Society of University Cartographers* 22(1): 17-25. ■