Cartography and Geographic Information Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcag19

Line Generalization Based on Analysis of Shape Characteristics

Zeshen Wang & Jean-Claude Müller

Published online: 14 Mar 2013.

Line Generalization Based on Analysis of Shape Characteristics

Zeshen Wang and Jean-Claude Müller

ABSTRACT. Many solutions for line generalizations have already been proposed. Most of them, however, are geometric solutions, not cartographic ones. The position we take in this paper is to observe school-case solutions available in standard cartographic books and try to replicate those automatically. A central criterion guiding the process of cartographic generalization is line structure, which itself can be decomposed into a series of line bends. Hence our solution is to preserve the overall structure with line bends which are mathematically defined according to size, shape, and context. Rules are subsequently applied using operators such as elimination, combination, and exaggeration. The algorithms that were used are both procedural and knowledge based. Various experiments were conducted on physical and political geographic lines, and we show the graphical results so that readers may visually assess them. Further research to improve the present solutions is discussed, particularly options for avoiding conflicts in large-scale reductions.

Introduction

Cartographic generalization is one of the most intellectually and technically challenging aspects in the mapping process (Müller 1991, McMaster and Shea 1992). This paper focuses on generalization of a single linear feature. In this area, many algorithms for automated generalization have been proposed and tested, but they still contain various deficiencies in reaching satisfactory results, especially for large-scale reduction (Monmonier 1986, Mark 1989, McMaster and Barnett 1993). Most of these techniques are based on mathematical procedures and deal with generalization as if it were exclusively a geometrically rooted problem, such as map projection and polygon overlay.

Many algorithmic approaches handle the line as a collection of points and measure point attributes such as angularity and distance as criterion for generalization, whereas cartographers detect the characteristics of a line (such as bends and relationship between bends) before undertaking a generalization operation. As Brassel and Weibel (1988) pointed out "mere succession of mechanical steps can suffice only in exceptional cases. What is needed is processing based on understanding."

Zeshen Wang is Senior Programmer at ESRI, 380 New York Street, Redlands, CA 92373. Tel: (909) 793-2853/ext. 2040. E-mail: <zwang@esri.com>. **Jean-Claude Müller** is Professor at the Geographisches Institut, Ruhr-Universität Bochum, Universitätsstr. 150, 44780 Bochum. Germany. Tel: 0234/700-3379; Fax: 0234/ 7094-180. E-mail:<jean-claude.mueller@rz.ruhr-uni-bochum. de>.

Rule-based systems have been proposed for generalization in a digital mapping environment (Mark 1991, Shea 1991, Müller et al. 1995) that aim to simulate manual generalization. Systems of this kind have been developed in such generalization areas as selection of features and depicting changes in symbology, but this approach cannot be adopted directly for line generalization. An example rule for point feature selection may indicate that symbols for cities that have population less than ten thousands should be eliminated. The premise for this rule can be easily detected when the population of cities is given. But rules followed by cartographers for line generalization are usually much more ambiguous.

What a cartographer perceives to be in a line, and how he interprets it before making a generalization decision, is not clearly understood (McMaster and Shea 1988). Experts in the field have different points of view. McMaster (1993) suggested: "Individuals seem to judge the shape of the line on two criteria: the directionality of the line and the basic sinuosity of the line." He devises a matrix to record direction and distance of each segment along a line and uses the multivariate technique of cluster analysis to classify lines into groups. In McMaster's experiment, lines collected from different natural features are classified into three groups: relatively smooth, quite sinuous, and multi-directional trend.

Buttenfield (1991) assumed that a line can be split into homogeneous pieces, thereby allowing better selection of algorithm and tolerance by bandwidth, concurrence, and other parameters. She believes that the relationships between line geometry and the relevant algorithms and tolerance can be represented by a set of rules. More measures for line geometry have recently been developed, and the relevant experiments according to these measures were carried out (Plazanet et al. 1995).

The method of line generalization presented in this paper is based on bend manipulation. It was first used in guidelines for manual generalization developed by the Swiss Society of Cartography (1977). An experiment in a digital environment following some of these guidelines was recently reported (Wang 1996). However, bends are not a feature like cities that can be retrieved from a data base. 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 (in the sense expressed by Brassel and Weibel 1988), i.e. detecting bends and computing their attributes on which generalization decisions can be made.

Plazanet (1995) did detect bends in her work and conducted some measurements on bends as well. Her purpose, however, was not to manipulate individual bends through such operations as elimination and combination. Rather, she uses detected bends and measures their attributes in order to segment a line and then apply an appropriate algorithm and tolerance to each section. Other approaches of this kind, though not directly applying bends for generalization, can be found in Perkal's algorithm (1966) using the Epsilon band and Li and Openshaw's algorithm (1992) based on a "natural principle" for objective generalization. The former method places a circle with diameter ε inside the region and rotates it in such a manner that it remains completely inside the area, while the later one defines a circle representing the so-called smallest visible object and forms a series of circles of same size, side by side, along the line. If a small bend falls totally inside a circle, the whole bend will be replaced by the center of the circle.

The method we propose is based on shape analysis; it uses information on the shape of bends along a line and their relationships to select appropriate generalization operations. Brassel and Weibel (1988) call the shape analysis a "structure recognition" which they consider the first step in their automated generalization framework. They further describe this process as aiming "at the identification of objects or aggregates, their spatial relations, and the establishment of measures of relative importance."

The following sections introduce bends and suggest their definition in a computer- understandable fashion. The next identifies the relevant attributes needed for making decisions in line generalization. We then describe generalization operators and their execution sequence in data processing. The final section evaluates the performance of the proposed prototype system through examples. The major advantages and disadvantages of the method are drawn in the conclusions, based on its principles and performance.

Basic Graphic Elements

Figure 1 presents the generalized versions of three line examples provided by Swiss Society of Cartography. Example (a) shows the elimination of small bends, whereas example (b) demonstrates the combination of bends on a contour line. From example (c) we see that a "natural" line does not become a geometrically straight line after manual generalization; it remains an irregular line, albeit less irregular than the original line. "Irregularities" on a line can be considered as small bends. Therefore, it is assumed that bends are the most fundamental constituting elements of a geographic line, and the generalization decisions made by cartographers will primarily depend on the attributes of the bends and their relationships.

Four rules for line generalization are listed below. The first three rules are derived through examination of the manual examples in Figure 1, and the fourth rule for bend exaggeration is derived from the general objective of preserving the characteristics and clarity of the original line when this is reduced at a smaller scale. The rules are:

1. Small bends and irregularities should be removed.
2. Two bends—if they are similar, next to each other, and their size lies under a given (yet to be defined) threshold—should be combined as one, while three bends can be represented by two.
3. Non-straight lines should not be replaced by geometrically straight lines.
4. An isolated bend, when its size is close to the above threshold, should be exaggerated.

These four rules do not exhaust the rules for manual line generalization, but they are fundamental in guiding the process of manual generalization. It is clear, from these rules, that manual generalization is not directly applied to each vertex of the line,

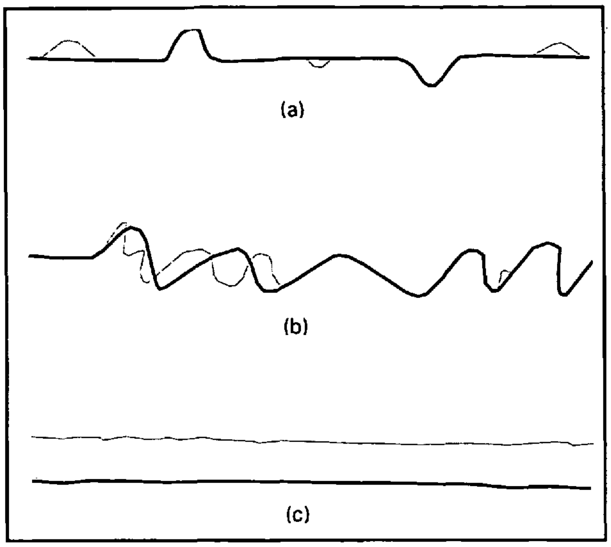

Figure 1. Manual generalization examples (after the Swiss Society of Cartography).

instead, it is applied to the bends that consist of numerous vertices.

Most existing line generalization solutions are based on geometric processing without previous shape analysis (this shape analysis is termed "structure recognition" in Brassel and Weibel (1988) and "cartometric evaluation" in McMaster and Shea (1992)). For example, the well known Douglas algorithm (Douglas and Peucker 1973) measures perpendicular distances of each intervening point as a unique criterion for point selection, the only operation the algorithm performs. As no bends are detected by the algorithm, the cartographic generalization rules based on bends cannot be followed by this algorithm.

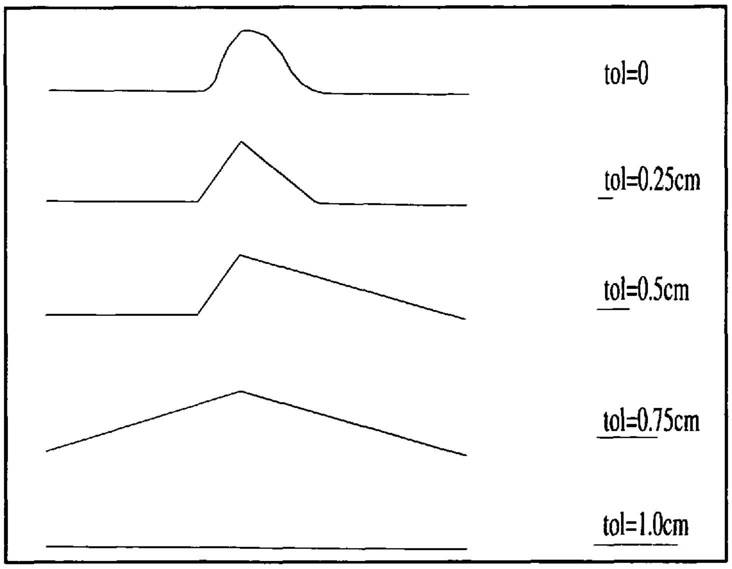

Figure 2. Applying Douglas's algorithm with increasing tolerance.

Figure 2 illustrates the generalization results of the Douglas algorithm applied on a simple line with one bend. The height of the bend is just under one centimeter, and five tolerances with increasing values are taken to test the algorithm. The change of the bend begins by a sharpening of the bend's angularity and leads to a distorted view which is eventually eliminated with the whole line being replaced by a geometrically straight line. It is obvious that the algorithm will also fail to carry the other proposed operations based on bend detection, i.e., bend combination and exaggeration.

Definition of a Bend

A basic assumption of this paper is that a line feature, such as a coastline, can be viewed as a sequence of bends. A human reader perceives two bends B1 and B2 in the line shown in Figure 3a. But these two bends are not explicitly indicated once they have been digitized, thus we need to find ways to define them so that computers can detect them from the XY string. A bend can be defined as that part of a line which contains a number of subsequent vertices, with the inflection angles on all vertices included in the bend being either positive or negative and the inflection of the bend's two end vertices being in opposite signs. Plazanet (1995) defined a bend in a similar way: bend is a fraction of a curve between two consecutive inflection points.

According to our definition, three bends of B1, B2 and B3 can be identified in the same line as shown in Figure 3c. The inflection of a vertex can be determined by the angular difference of the direction of the next segment minus the direction of the last segment at this vertex. Counterclockwise angles are measured positively as the line is digitized from left to right. As a result of this definition, the inflection angles in bend B1 and B3 are all negative, and those in B2 are positive. This definition brings up two additional characteristics of a geographic line:

- Positive bends and negative bends are always next to each other (an obvious characteristic).
- Each bend is next to another, and covers every vertex along the whole curve.

Figure 4 shows the bends of a meandering river that were extracted using the method defined above. Once the bends are determined, the attributes of each bend can be computed. Furthermore,

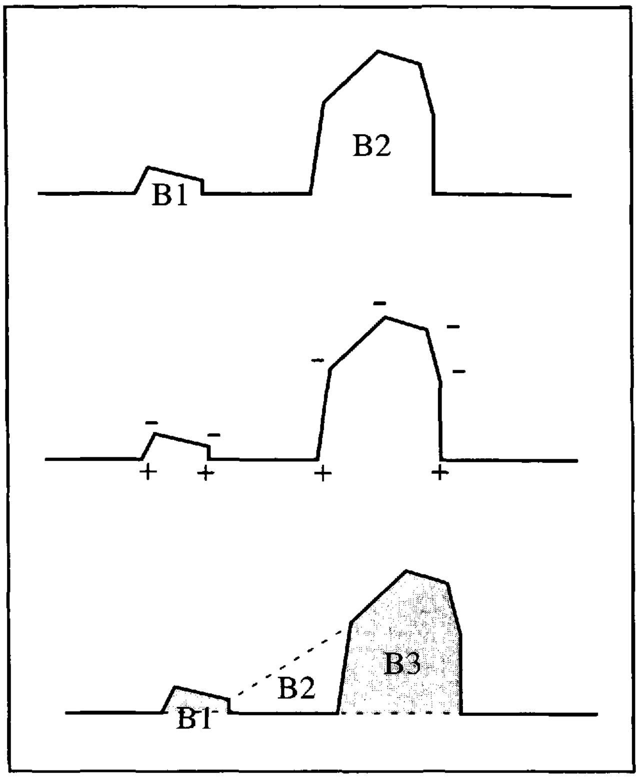

Figure 3. a) People perceive two bends. b) The signs of inflection on each vertex. c) Computer may detect three bends.

relationships such as similarity of neighborhood between bends can be worked out by comparing the relevant attributes.

Though the bends extracted by computer in Figure 4 look fairly similar to what people would see, some exceptional cases may occur and must be identified. These are described in the following subsections.

Gentle Inflection at End of a Bend

As mentioned earlier, a bend is delimited by the changing direction of inflection angles. But if the inflection that marks the end of a bend is quite small, people would not recognize this as the end point of a bend. The end points should be moved outward only when the inflection angle is small and the new baseline is shorter than the old one. In Figure 5 for example, A and C are two ends of a bend and AC is its baseline. Since the inflection in A is small and line BC is shorter than AC, the end point is corrected to B.

Self-line Crossing When Cutting a Bend

When the sum of inflections for all vertices inside a bend is large, generalization may lead to self-line crossing. For example, in Figure 6, CI is a baseline of a bend, and the sum of inflection angles from point D to H is about three right angles, i.e., 270 degrees. When a self-line crossing is detected in one end, the end point of the bend must be moved outward until no self-line crossing can be generated. In the example shown in Figure 6, self-line crossing occurs around the end point C, therefore the end point should be moved outward to A. As in the preceding example, the shortest neck becomes the new baseline which is then corrected to AI.

Situation Assessment

The premise of the four rules mentioned above involves the attributes of a single bend or the relationships between bends. The size of a bend is certainly the most important factor to be considered in geometric generalization. The shape of a bend is another attribute that can significantly affect the outcome of the generalization process. In this paper, however, only the relationship of neighboring bends are considered.

Attributes of a Single Bend

Some line generalization decisions (e.g., elimination) are based primarily on the attributes of a single bend.

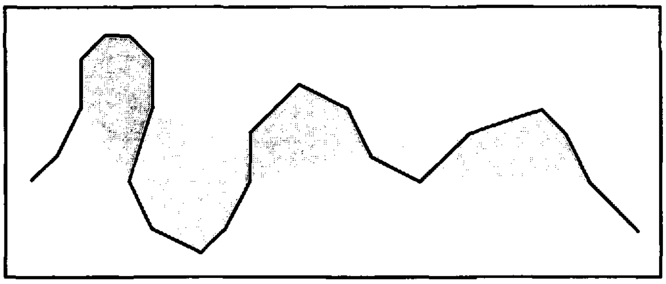

Figure 4. Bends detected along a meandering river.

Size of a Bend
The most sensitive attribute affecting bend selection/elimination is size. The size of a bend is defined as the area of the polygon enclosed by the bend and its baseline (see the shaded areas in Figure 4).

Shape of a Bend
In this paper it is assumed that a bend can only contain consecutive vertices that have the same direction of inflection. Thus the polygon formed by the bend

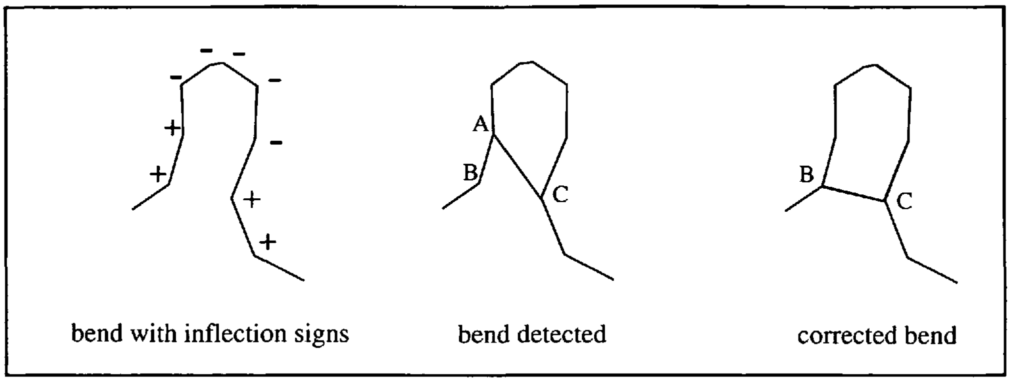

Figure 5. Bend end defined by a gentle inflection may need correction.

and its baseline will usually be a simple polygon type, e.g., a convex polygon. Figure 7 presents some simplified shapes of bends, all having the same size (defined by the area of the corresponding polygons). Elongated and flat bends have a greater chance of being retained than do circle-shaped bends. This is because elongated bends are more prominent and flat bends cause less conflict when scale reduction takes place.

The shape of a bend polygon can be described by a compactness index (cmp) which, in turn, is defined as the ratio of the area of the polygon over the circle whose circumference length is the same as the length of the circumference of the polygon. Bend C is a rectangle-shape bend with a baseline twice as long as its height. Bend D has exactly the same shape as bend C but it is elongated in the vertical direction. Both bends have the same compactness index. Bend E has the smallest compactness index. To simplify the criterion for bend selection/elimination, the inverse of the compactness index is used to adjust the bend size, and this adjusted size is then used as primary criterion for bend selection:

$$\text{adjusted size} = \text{area} * (0.75/\text{cmp}) \qquad (1)$$

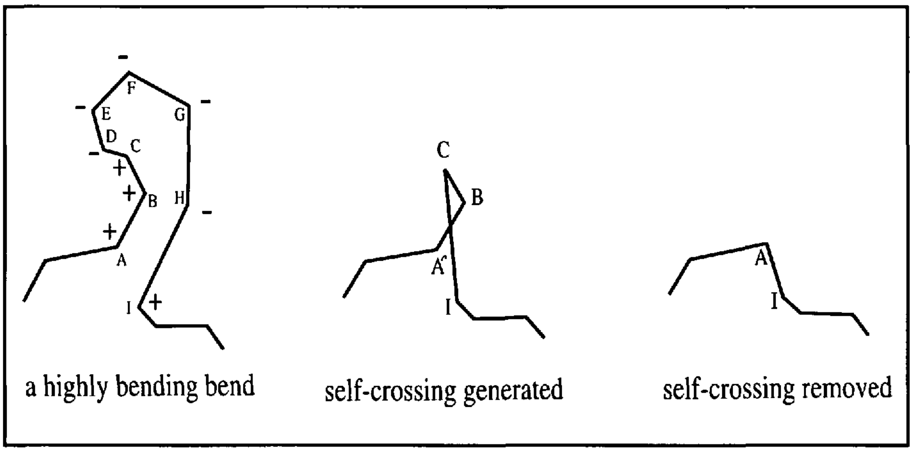

Figure 6. Self-line crossing generated by bend-cutting should be corrected.

As circle-shaped bends are rare in reality, half-circle bends with a compactness index of 0.75 are used here as the standard shape. The size adjustment factor is set to 0.75 over the compactness index, leading to the following modification:

$$\text{adjusted size} = \text{area} * (0.75/\text{cmp}) \qquad (2)$$

According to this formula, the adjusted size of most bends is not equal to their actual area, except when a bend is a half-circle whose adjusted size is the same as its actual area. The size of a bend will be adjusted down if its compactness is >0.75 (e.g., a circle-shaped bend with a cmp value 1), otherwise the size will be adjusted up. For example, if the area of a circle-shaped bend is A, and the compactness index of a circle is 1, the adjusted size will be reduced from A to 0.75A by using the above formula with substitution values for area and compactness. Rule three (i.e., that a non-straight line should not be replaced by a geometrically straight line) is realizable because a flat bend has very small compactness index, and the adjusted size of the bend will become larger than the user-set threshold for bend elimination. Thus, a very flat bend with area A and compactness index 0.01 will have its adjusted area increasing from A to 75 times of A, which will likely be greater than the user-defined tolerance, saving the bend from elimination.

The Context of a Bend

Some generalization operations, such as bend exaggeration and combination, depend on the relationship of a bend with its neighboring bends.

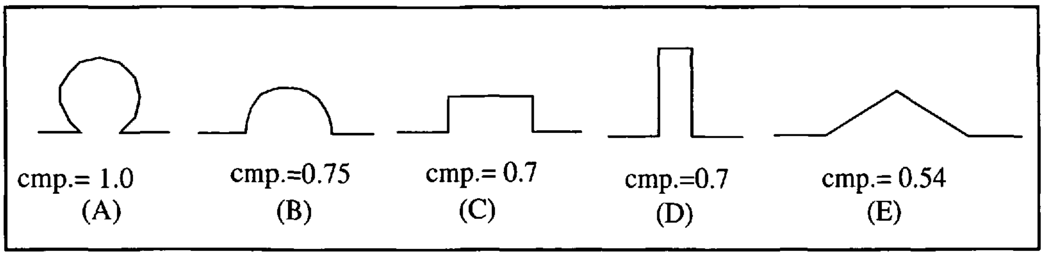

Figure 7. Five bends with same sizes but different compactness.

Isolated Bend

If it is true that bends are next to each other and cover every vertex along the line, then "isolated" bends can never be found. However, if the extended curves on both sides of a bend are smooth and longer than a given length, the bend can be defined as an isolated bend. Average curvature, which is geometrically defined as the ratio of inflection over the length of a curve, can be used to describe the bend itself and its extension on both sides. When the average curvatures on both sides of the bend are smaller than a threshold value and also much smaller than the average curvature of the bend itself, then the bend is considered to be an isolated bend.

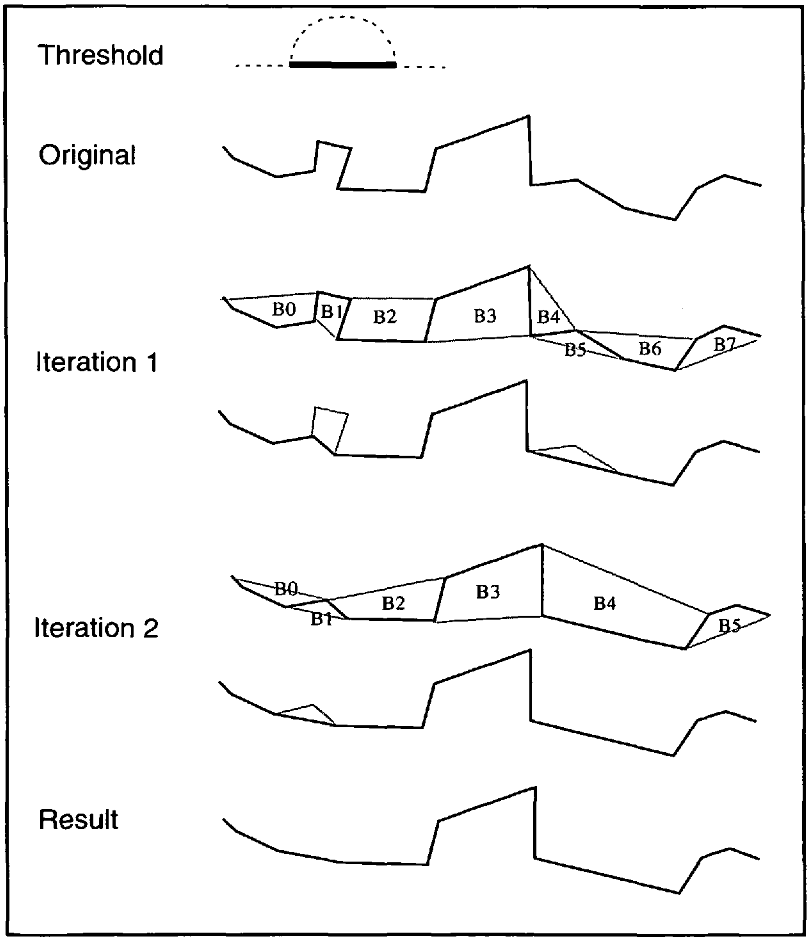

Figure 8. Bend elimination through iteration.

Similar Bends

Geometrically, similar polygons are identical in shape no matter how different they are in size. However, in the context of line generalization, bends are considered similar if both their size and shape are fairly close, i.e., only bends similar in both size and shape are comparable. Size comparison is simple, but shape similarity is harder to define as this involves subjective judgment.

In this paper, the shape of a bend is defined by the compactness index which does not differentiate between a flat and an elongated bend. From Figure 7, it is obvious that the bends C and D are not similar. To differentiate between these two types of bends, the length of their baseline is added as a measuring component of the similarity. The components (size, compactness index, and baseline) can be mapped along three orthogonal axes, such that each bend is represented by one point in this 3-dimensional space. The Euclidean distance between two bends is then calculated to differentiate between two bends. As this distance depends on the measure being used (e.g., mile, kilometer), normalization is carried out by dividing each measured length by an average. For example, if bend 1 has four unit areas and bend 2 has six unit areas, the average size is five units, and the normalized areas of bends 1 and 2 are 4/5 = 0.8 and 6/5 = 1.2, respectively.

Operators and Implementation

In order to carry out bend generalization, a prototype system has been developed. Before using the system, the user must set a minimum diameter for a half-circle bend, and this minimum will be used as tolerance and reference for bend elimination and other bend operations.

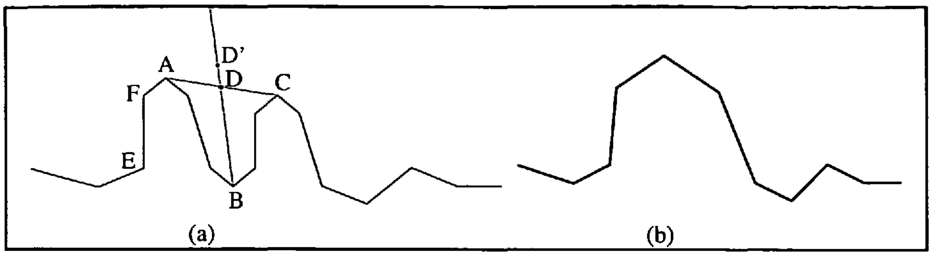

Figure 9. Bend combination.

Elimination Operator

Bend elimination is done by replacing the bend curve by its baseline. Figure 8 demonstrates how the small bends are cut. Since the baselines of consecutive bends are not connected, the cutting action must be executed iteratively, by removing local minimal bends in each loop. A bend that is smaller than either of its two neighboring bends is defined as a local minimal bend. In the first iteration, bends 1 and 5 are local minimal bends, and they are eliminated. Note that the first and the last along the line are bends 0 and 7, although these two bends do not fulfill the bend definition because they are open end. The two bends at the ends of a line are always assumed to have bigger size than their neighboring bends. Once a local minimal bend is removed, a bigger bend forms around it, and this new bend may be large enough to avoid being generalized. In our example in Figure 8 only four bends are left after iteration.

The iteration for generalization is carried out until every bend on the line is larger than the area threshold. A cartographer may accomplish generalization in one swoop of the pen because he/she can perceive and interpret both unwanted detail and important characteristics at the same time, but our prototype system can only do bottom-up processing, starting from cutting smaller bends.

Combination Operator

As Müller et al. (1995) pointed out, combination is one of the least developed operations in automated line generalization. Figure 9 shows three similar consecutive bends, and the goal of generalization is to combine the first and the third bends as one. The operation is carried out as follows.

First, the peak of each bend is found by comparing the sum of the distance from each vertex to the two end points of the bend, and the vertex which has the largest sum is defined as the peak of the bend. In our example there are three peaks labeled A, B, and C. Point D is the center of line AC, and point D' is the peak of the combined bend. Peak D' is situated on the extension of line BD. BD'/BD is set slightly greater than 1.0 so as to be able to enlarge the height of the combined bend and reducing its distortion. In our case, the ratio was arbitrarily set to 1.2, although other values may be selected by the user.

Second, the left half of bend 1 and the right half of bend 3 should be moved toward the new peak D.' Point E is the first node of bend 1, so it should stay in its place. Point A is the peak of bend 1, and its moving vector is AD.' The vertex between A and E, i.e. point F, should move in the same direction as vector AD'; its magnitude is proportional to FE/(AF+FE). The resulting line is shown in Figure 9b.

Cartographers are able to conduct a more complicated operation, such as combining three bends as two. However, our prototype system can only combine two similar bends as one. In a long series of bends, the program will handle each set of two bends recursively and eliminate the last bend if the bend number is odd.

Exaggeration Operator

Exaggeration is often used to prevent spatial conflict and emphasize the importance of a feature. In line generalization, conflict may occur when the scale is reduced. Mackaness and Fisher (1987) suggested two algorithmic solutions—proportional radial enlargement and the use of Gaussian distributions to determine the displacement. The former approach involves selecting a center and moving all the points away from the center by a distance d which is proportional to the point's original distance from the center.

The advantage of this method is that the shape of the displaced feature will be preserved. In our case, however, the bend subjected to exaggeration is not an isolated feature; rather, it is a part of a long curve, which makes it difficult to decide where to stop the displacement.

The second method using Gaussian distributions also needs to find a center point for displacement but instead of moving the outer points

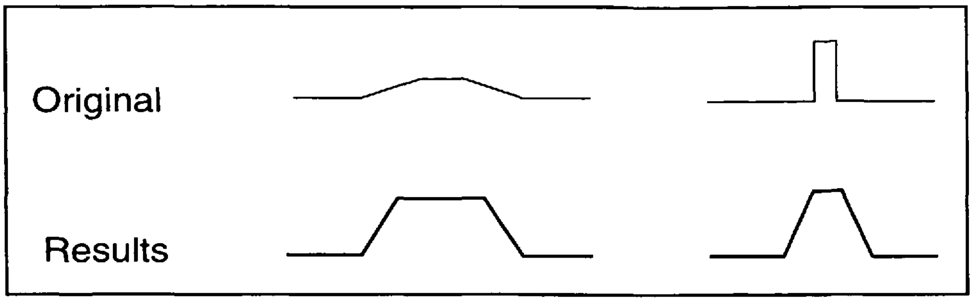

Figure 10. Exaggeration using Gaussian distributions causes distortion.

farther, the distance of displacement decays gradually from the center to the fringe according to a given Gaussian distribution. This method was applied in Figure 10, where the center of a bend's baseline is the center for displacement. The disadvantage of this method is that it may introduce a distortion in the bend, such that a flat bend will become taller and a narrow bend will be wider (Figure 10).

Program Implementation

The original program was written in PROLOG language that is specifically designed for expert system development. PROLOG's power lies in its ability to infer facts from other known facts. If all the bends and their attributes along a line are known facts, the appropriate operator can be found through symbolic reasoning. However, the language is much weaker in numerical and computational analysis than are such procedural languages as C and FORTRAN, and the user, consequently, has to prepare all the information manually in order to use the rules. For instance, the position of each bend must be marked and its attribute must be calculated by the user. Then, using PROLOG's inference engine these input facts can be examined on the premise of each rule. This manual way of collecting information is obviously not practical for an operational system. Moreover the generalization operations also require intensive numeric manipulations.

Given these shortcomings, the program was rewritten in FORTRAN. The FORTRAN version can handle bend detection and analysis, as well as the selection of generalization operators and the actual generalization operations. In the final code, the three rules, each equipped with its operators, are written as conditional statements.

Note that among the three rules used in this system, the application of the exaggeration and combination rules requires stricter conditions than the elimination rule. Suppose a small bend appears along a line, and the size of the bend is under the threshold, two rules, i.e. elimination rule and exaggeration rule can be invoked. When this kind of conflict occurs, a simple solution is to give the rule with more constraints higher priority (Fu et al. 1987). In our prototype system, the priority consideration was realized by arranging three conditional statements as follows:

- Looking for isolated bends for exaggeration;
- Checking for the presence of similar bends for combination; and
- Searching locally for smaller bends for elimination.

The number of loops performed going iteratively through the line depends on the user-defined tolerance and the input data. If the input line is very detailed and the tolerance is large, it may take 20 loops to complete the generalization operation. This bottom-up iteration strategy and its complex bend-detection and geometric manipulation slows down the generalization process. A flowchart of the steps involved is shown in Figure 11.

A rule-based system, according to Turban (1992) is "a system in which knowledge is represented completely in terms of rules." In our case, we used rules derived from knowledge about generalization, and we searched for geometric solutions in order to satisfy those rules. We used only a few rules, recording the symbolic matching and reasoning of the original rules in conditional statements that mainly deal with numerical comparisons. Hence technically speaking, our prototype system cannot be called a rule-based system. Basically, our algorithm provides a geometric and procedural solution, but, unlike some previous algorithms (e.g., the Douglas algorithm), it is directly derived from generalization knowledge and rules.

Results of Experiments

We experimented with the line generalization prototype system on several types of line works (Figure 12). For single lines, the scale of the generalized results was as small as 5 percent of the original. For area features with linear boundaries, however, the scale reduction could not be as large

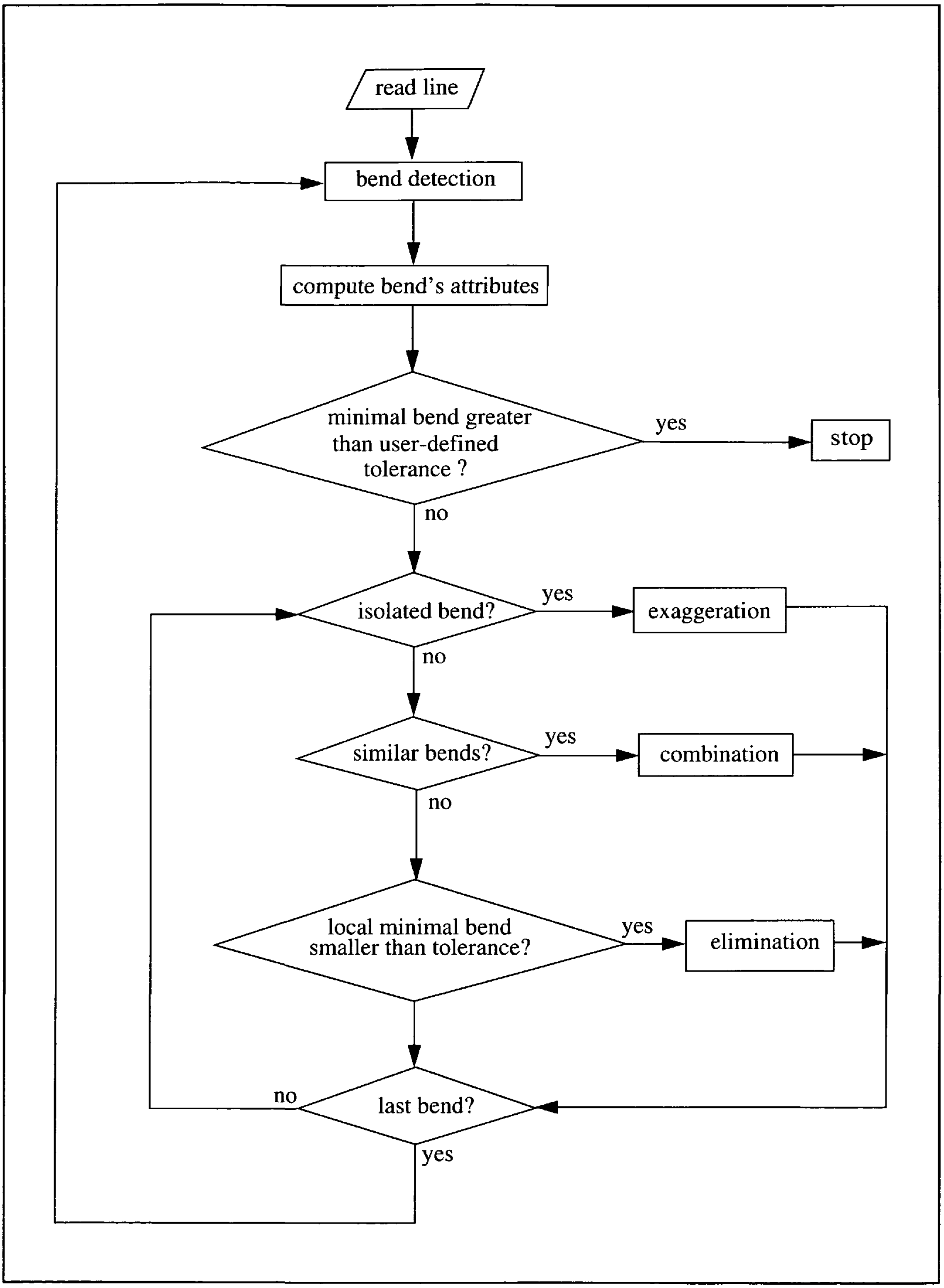

Figure 11. Flow chart of the prototype system.

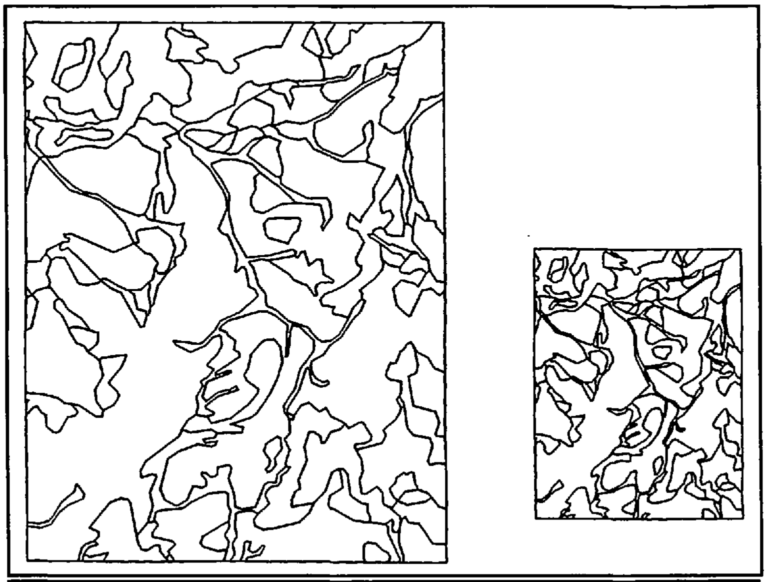

Figure 12.1. A soil map.

because a large reduction produces spatial conflicts in each area (a solution to this problem has yet to be found). Another problem can be observed when applying this method to polygons: a node may have three or more lines joining to it. After some bends near a node have been cut, crossing between lines may occur. The third problem is memory consumption. The proposed method needs more memory space to keep the generalized line because we want to preserve the round bends, not distort them. This problem may be solved by applying the Douglas algorithm as a post-process to each retained bend using small portions of the bend's amplitude as tolerance so that the bend would not be simplified into a straight line. All the single line generalization examples given below use comparable tolerance levels and feature selection.

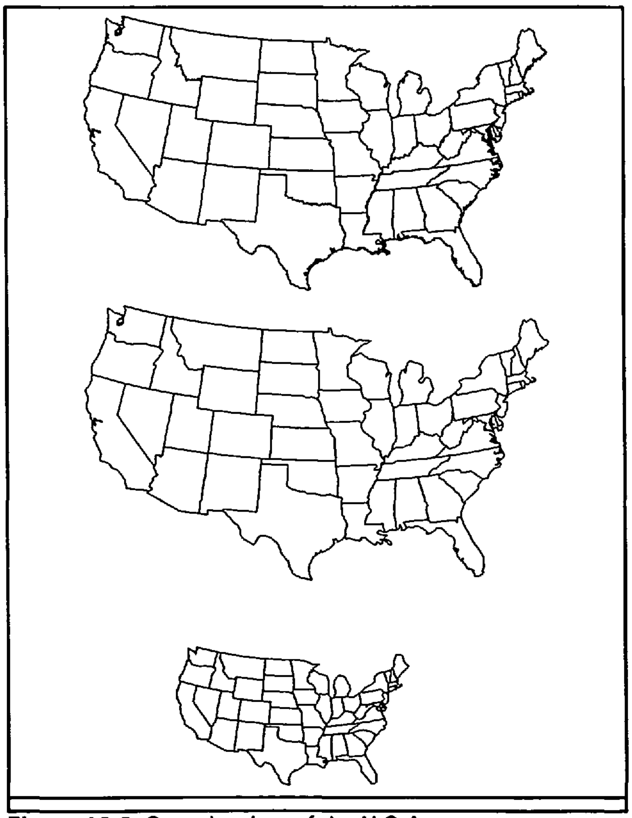

Figure 12.2. State borders of the U.S.A.

Visual examination can be used here to assess the performance of the proposed bend-based algorithm. Further assessments could be conducted using numerical analysis. Several numerical measures, including line sinuosity and line displacement, have previously been suggested (McMaster 1983). The following examples were used:

- A soil map (Figure 12.1).
- State borders of the U.S. (Figure 12.2). Because the original data (top) for the coastlines of Texas, Florida, Virginia, North Carolina and Washington State are too detailed to be presented in the limited space, the prototype system is used to generate a clean map (middle) in the same scale, and the clean map is further generalized to produce the final smaller-scale map (bottom).
- The coastline of Banks, British Columbia (Figure 12.3); comparing results with the Douglas algorithm (top).
- Coastline generalization (Figure 12.4); comparing manual products (top, extracted from the Swiss Society of Cartography 1977) with the results of the Douglas algorithm (middle) and our approach (bottom).
- Lake shore line generalization (Figure 12.5); comparing results with the Douglas algorithm.
- Isolated bend exaggeration (Figure 12.6).

Conclusions

Unlike with most existing line generalization algorithms, our algorithm is based on knowledge in the cartographic domain. The previous algorithms may be translated into rule-like statements; for instance, the Douglas algorithm can be described as "IF the perpendicular distance of intervening points of a line segment is smaller than tolerance, THEN these points should be eliminated." However this "rule" is not based on cartographic knowledge, and not even followed by many cartographers.

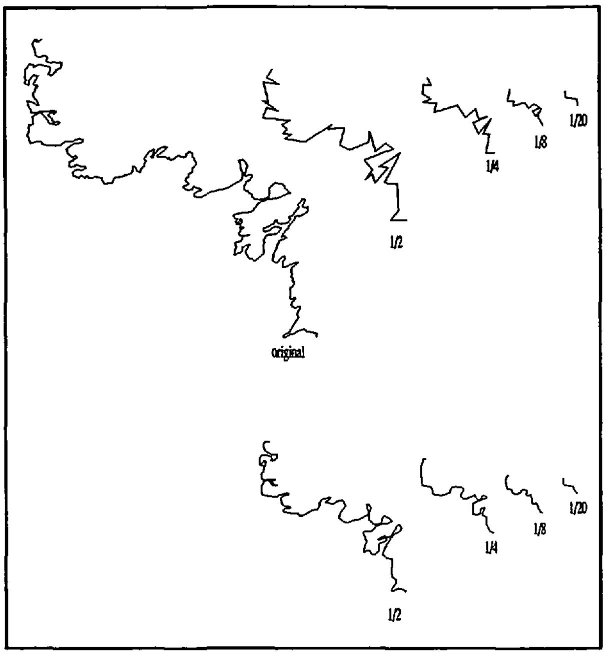

Figure 12.3. The coastline of Banks, British Colombia.

The philosophy behind our approach is simple: if we want a computerized system to work the same way as does a cartographer, we must find out what practical rules cartographers follow, and try to encode these rules into a computer program. Accordingly, shape analysis is conducted to extract graphic elements and evaluate their attributes.

The extracted graphic elements must carry high-level information of the line. Our experimental system extracts bends as graphic elements. A bend has several attributes, including size, which may be small or large, and shape, which is defined by a compact index and the similarity with other bends. Once the bends are identified and their attributes and contexts are evaluated, the appropriate operation (bend elimination, combination, or exaggeration, and prevention of straight line effects) can be determined. Unlike the measures and parameters used in other algorithms, however, the bend attributes we use are subject to interpretation, i.e., may be viewed differently from person to person. For example, where a bend starts and ends, and what are the criteria to define similar bends are both subject to the viewer's perception.

The main points of departure between the proposed line generalization method and other published algorithms are as follows:

- Application of rules which are directly derived from manual generalization knowledge.
- Selection of a generalization operation on the basis of an analysis of the shapes of bends and its application to the graphic elements of the bends. In contrast, other algorithms usually work with information related to individual points.
- Integration of numerous operations (e.g., elimination, exaggeration and combination) into a single program. An algorithm-based solution is usually equipped with a single operation, e.g. smoothing routines can only smooth a line.

As a consequence of points one and two, large reduction rates become possible. For extreme reduction rates, corrective actions can be

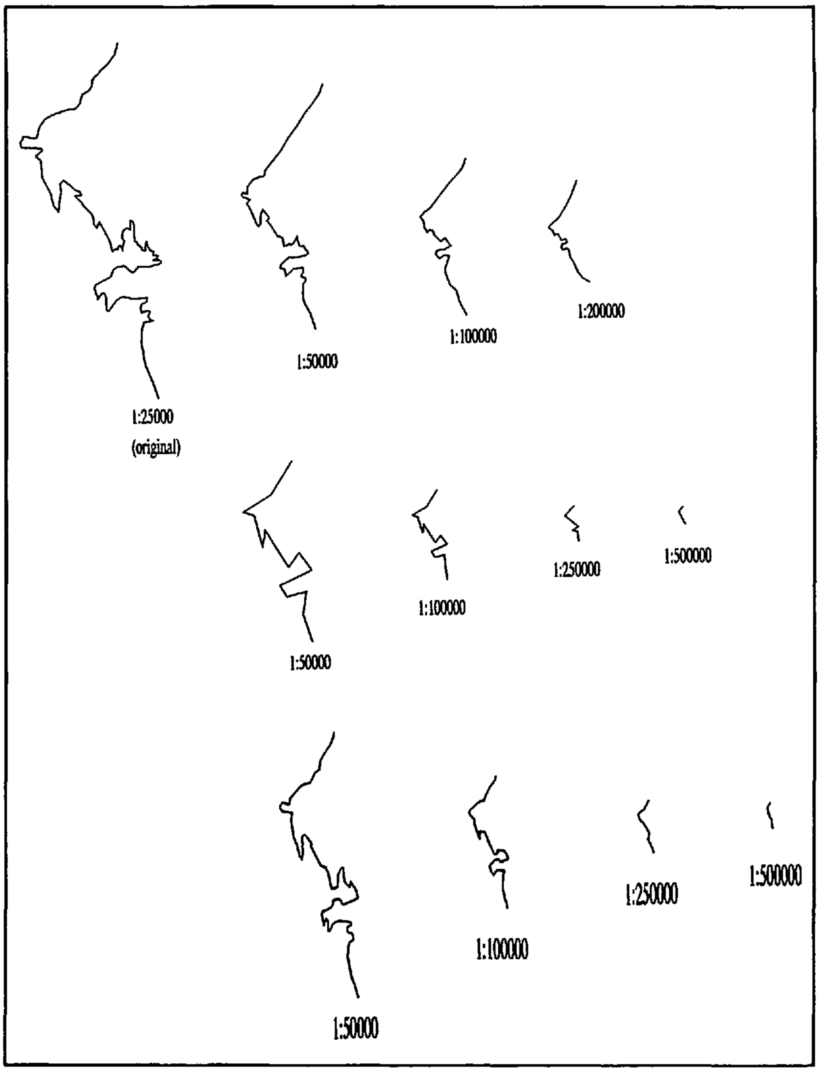

Figure 12.4. Coastline generalization.

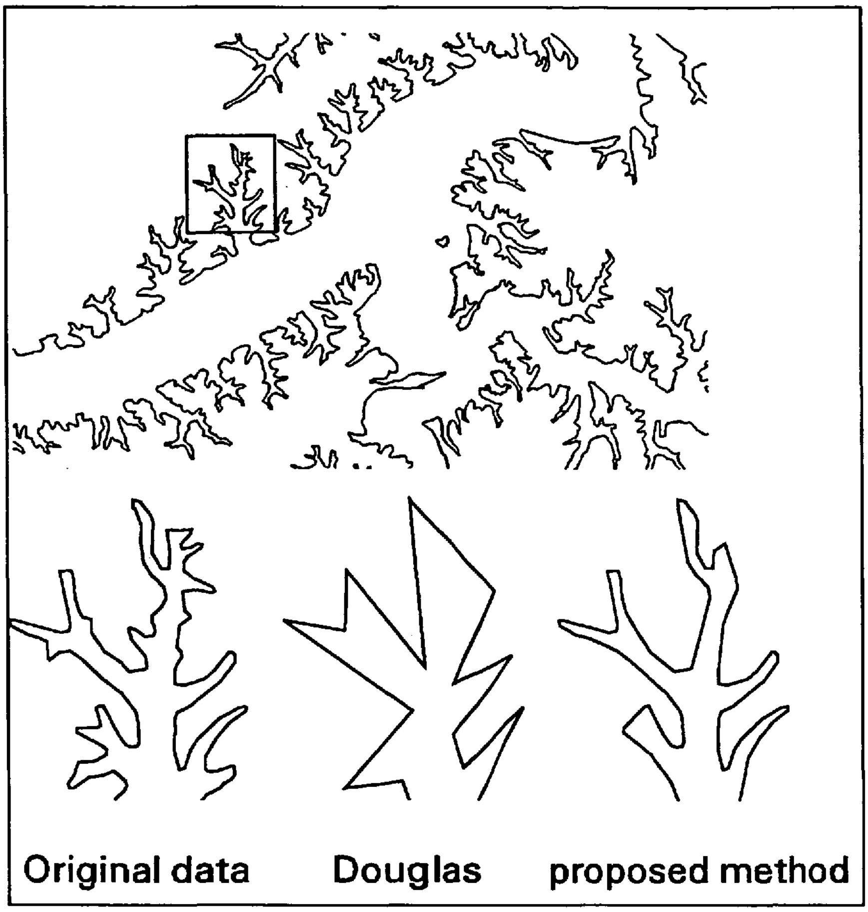

Figure 12.5. Lake shoreline generalization.

applied to avoid topological distortions (such as self-line crossing in the Douglas algorithm).

With bend-detection capability, one additional advantage may be gained. Generalization often has a tendency to emphasize human activity. For coastline generalization, for example, cartographers usually want to keep or exaggerate peninsulas rather than bays. According to the definition of a bend used in this paper, all peninsulas are either positive or negative, depending on the direction of digitization. By assigning an extra percentage to the size adjustment formula, the peninsulas will have more chance to be preserved.

Figure 12.6. Isolated bend exaggeration.

The experimental system described here 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. The generalization solution presented by Wang and Müller (1993) could be integrated into this system.

As earlier mentioned, a solution will have to be found for large-scale reduction of area features with linear boundaries, without producing spatial conflicts in each area. Moreover, the proposed method, just like other line generalization approaches, does not move the ends of line, even when the end nodes are an intervening point of a small bend. The problem of different line geometry need further research.

There is also need for additional cartographic rules for line structure recognition to enable more sophisticated operations. A twisting mountain road with many bends is interpreted by cartographers as a winding road, but our prototype system can check only a few bends at a time.

REFERENCES

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

Buttenfield, B.P. 1991. A rule for describing line feature geometry. In: Buttenfield, B.P., and R.B. McMaster (eds), *Map generalization: Making rules for knowledge representation.* London: Longman, pp. 150-71.

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

Fu, K. S., Z. X. Ci, and G. Y. Xu. 1987. *Artificial intelligence and its applications*. Beijing: Press of Tsinghua University.
Li, Z., and S. Openshaw. 1992. Algorithms for automated line generalization based on a natural principle of objective generalization. *Int. J. Geographical Information System* 6(5): 373-89.
Mackaness, W.A, and P.F. Fisher. 1987. Automatic recognition and resolution of spatial conflicts in cartographic symbolization. In: *Proceedings of AutoCarto 8*, Baltimore, Maryland. pp. 709-18.
Mark, D. 1989. Conceptual basis for geographic line generalization. *Proceedings of AutoCarto 9*, Baltimore, Maryland. pp. 68-77.
Mark, D. 1991. Object modeling and phenomenon-based generalization. In: Buttenfield, B.P., and R.B. McMaster (eds), *Map generalization: Making rules for knowledge representation*. pp. 103-18.
McMaster, R. 1983. A Quantitative analysis of mathematical measures in linear simplification. Unpublished Ph.D. dissertation, Department of Geography-Meteorology, University of Kansas.
McMaster, R. 1993. Knowledge acquisition for cartographic generalization: Experimental methods. In: Müller, J-C., J.P. Lagrange, and R. Weibel (eds), *GIS and Generalization: Methodological and Practical Issues*. London: Taylor & Francis, pp. 161-79.
McMaster, R., and L. Barnett. 1993. A spatial-object level organization of transformations for cartographic generalization. In: *Proceedings of Auto Carto 11*, Minneapolis, Minnesota. pp. 386-95.
McMaster, R., and K.S. Shea. 1988. Cartographic generalization in a digital environment: A framework for implementation in a geographic information system. In: *Proceedings of GIS/LIS '88*, San Antonio, Texas. pp. 240-249.
McMaster, R. and K.S. Shea. 1992. Generalization in digital cartography. Publication supported by the A.A.G., Washington, D.C. 134 pp.
Monmonier, M 1986. Towards a practical model of cartographic generalization. In: *Proceedings of AutoCarto VII*, London, U.K. pp. 257-66.
Müller, J-C. 1991. Generalization of spatial databases. In: D.J. Maguire, M.F. Goodchild, and D.W. Rhind (eds), *Geographic information systems*. London: Longman Ltd, pp. 457-75.
Müller, J-C., R. Weibel, J.P. Lagrange, and F. Salge. 1995. Generalization: State of the art and issues. Müller, J-C, J.P. Lagrange, and R. Weibel (eds), *GIS and generalization*. London: Taylor & Francis, pp. 3-17.
Perkal, J. 1966. *An attempt at objective generalization*. Discussion Paper No. 10. Ann Arbor, Mich: Michigan Inter-University Community of Mathematical Geographers. Paper translated by W. Jackowski from Julian.
Plazanet, C. 1995. Measurement, characterization and classification for line generalization. In: *Proceedings of AutoCarto 12*, Bethesda, Md., pp. 59-68.
Plazanet, C., J-G. Affholder, and E. Fritsch. 1995. The importance of geometric modeling in linear feature generalization. *Cartography and Geographic Information Systems* 22(4): 291-305.
Shea, K.S. 1991. Design considerations for an artificially intelligent system. In: Buttenfield, B.P., and R.B. McMaster (eds), *Map generalization: Making rules for knowledge representation*. London: Longman. pp. 3-20.
Swiss Society of Cartography. 1977. *Cartographic generalization*. Cartographic Publication Series No. 2. English translation by Allan Brown and Arie Kers, ITC Cartography Department, Enschede, The Netherlands.
Turban, E. 1992. *Expert systems and applied artificial intelligence*. New York: Macmillan Publishing Company. p. 797.
Wang, Z. 1996. Manual versus automated line generalization. In: *Proceedings of GIS/LIS '96*, Denver, Colorado, pp. 94-106.
Wang, Z., and J-C. Müller. 1993. Complex coastline generalization. *Cartography and Geographic Information Systems* 20(2): 96-106. ▪