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

Hierarchical Methods of Line Simplification

Robert G. Cromley

Published online: 14 Mar 2013.

To cite this article: Robert G. Cromley (1991) Hierarchical Methods of Line Simplification, Cartography and Geographic Information Systems, 18:2, 125-131

To link to this article: http://dx.doi.org/10.1559/152304091783805563

Technical Note

Hierarchical Methods of Line Simplification

Robert G. Cromley

ABSTRACT. *An important preprocessing operator in a digital mapping system is line simplification. A line-simplification operator generalizes digital, geographic base files to ensure that the amount of information to be displayed played matches the geographic scale at which it is displayed. Many different simplification algorithms have been proposed to accomplish this task. Each algorithm has an implicit objective regarding which points to retain when caricaturizing a line. Each caricature is dependent on a tolerance level specified at the outset. This paper proposes a hierarchical approach to line simplification. Hierarchical classification methods are frequently used to group objects by their thematic attributes. A similar method is now used for the parallel problem of line simplification. The advantage of this approach is that a set of line caricatures corresponding to alternate tolerance levels can be calculated and compactly stored in tree or list structures.*

KEYWORDS: *line simplification, hierarchical grouping, cartographic generalization.*

Introduction

The electronic encoding of cartographic data has made it possible to record far more data than are necessary to represent digital lines. Methods for performing automated line generalization have become a critical component of systems for processing digital map data. Numerous algorithms have been developed for solving the line simplification problem (Zycor 1984 and McMaster 1987a). A common characteristic of most simplification routines is that the reduced version of a line is generalized by retaining only the "characteristic" points of the string. These characteristic points should include endpoints, inflection points, and local maxima and minima (Freeman 1978). Most frequently, a prespecified bandwidth tolerance is superimposed on the original string, such that the perpendicular distance between each eliminated point and the resulting simplification is within the tolerance [Douglas and Peucker (now Poiker) 1973; Peucker 1976]. Optimal procedures have been developed to find the minimal subset of points that satisfies any bandwidth tolerance (Cromley and Campbell 1991).

Most of these simplification algorithms have been evaluated by a variety of error measurements based on certain geometric properties of a line, including the length of the simplified string, its total vector displacement and areal displacement from the original string, and its angularity (Figure 1) (McMaster 1983, 1986, 1987b; White 1985). A vertex substitution algorithm has also been developed to optimize a given geometric property for a prespecified number of retained points (Cromley 1988).

In all cases, however, the simplification routines are oriented to produce a single solution corresponding to one desired compilation. In a map-processing system, it may be desirable to have the ability to continuously recompile the geographic data base to cor-

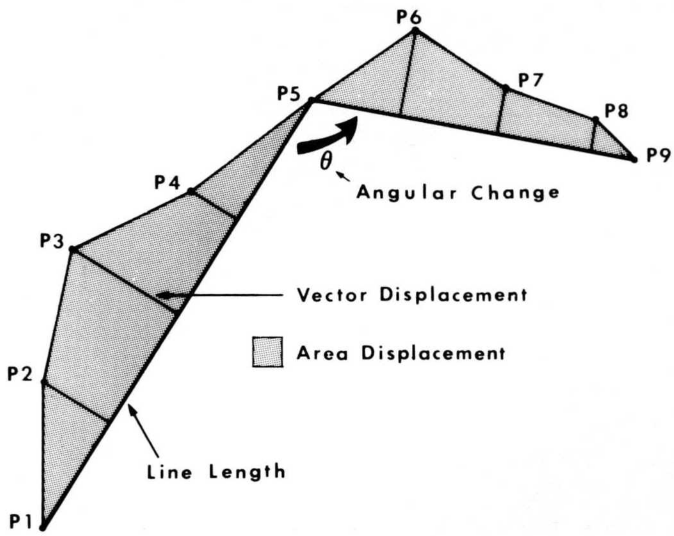


Figure 1. Alternate geometric measures of simplified line quality.

Robert G. Cromley is affiliated with the Department of Geography, University of Connecticut, Storrs, CT 06269.

respond to different scale representations without having to start from scratch each time. Douglas and Peucker (1973) developed a hierarchical procedure for line simplification, but did not exploit the special tree structure for storing and retrieving hierarchical information. The techniques presented here build on the work of Jones and Abraham (1986, 1987) for the design of scale-independent cartographic data bases. The purpose of this paper is to present general hierarchical methods for constructing a simplification tree from which any desired scale representation can be efficiently retrieved.

Hierarchical Classification Methods

Classification methods are commonly used in cartography to reduce the complexity of a thematic attribute distribution, while maintaining the salient elements of the distribution subject to a set of controls. The problem is one of removing the "noise" from the distribution by grouping like values into homogeneous classes. The control frequently associated with any given classification is the number of classes; the fewer the number of classes, the more generalized is the resultant classification. Hierarchical classification imposes the further control that members of the same group in an N-group classification must be members of the same group in an (N-1)-group classification. For example, in Figure 2a, the objects having values of 1 and 6 are part of the same group in a four-group classification. In a five-group classification, they still belong to the same group; they will remain in the same group all the way to a single-group classification.

There are two basic approaches to hierarchical classification: divisive methods and agglomerative methods (Burrough 1986, chapter 7; Johnston 1976). Divisive methods (the top-down approach) begin by assuming that all objects are members of one group; at each stage of the classification, this larger group is divided into successively smaller subgroups until there are as many groups as unique data values, and each value is its own group. This process can be represented by a tree diagram (Figure 2b) in which the terminal nodes at the bottom of the tree are single-value groups that correspond to the individual data values, and nodes higher up the tree represent groups that can be split into two sub-groups. At each node of the tree, some measure of similarity is used to determine how the objects will be subdivided into two lower-order groups.

On the other hand, agglomerative methods (the bottom-up approach) start with each data value as a separate group. At each stage of the classification, two smaller groups are merged together into a larger group, based on a similarity measure that states that the groups being merged are more alike than any other potential pair of groups (Figure 2a). Both divisive and agglomerative methods attempt to optimize the amount of intra-group similarity and/or maximize the amount of inter-group dissimilarity at each stage; however, because they are restricted by the hierarchical control that members stay in the same group, they cannot produce the "best" classification with respect to a given number of classes for any objective. See Jenks and Caspall (1971) for a discussion of optimal non-hierarchical data classification for choroplethic map display.

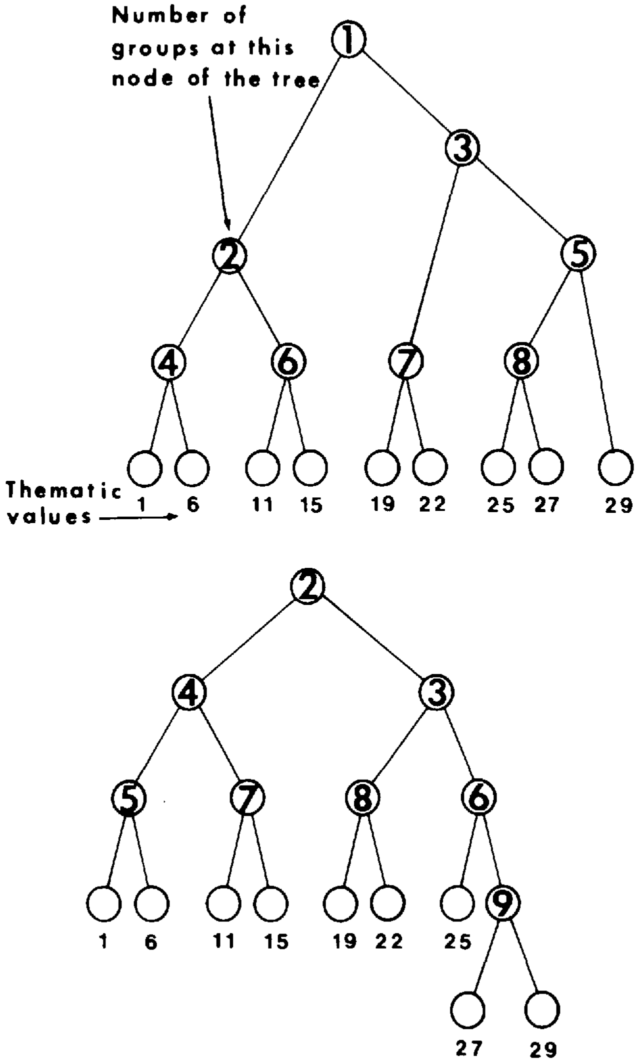


Figure 2. Types of hierarchical groupings. The thematic attribute for each object is given below its vertex. (top) Bottom-up. The number inside each tree vertex is the number of groups at that level of classification. Agglomerative methods start with each object as a separate group, N=9 in this case. (bottom) Top-down. The number inside each tree vertex is the number of groups at that level of classification. Divisive methods start with only one group and each object is a member of it.

Hierarchical Line Simplification

Line simplification is also a problem of removing the "noise" from a distribution, although the objects are line segments rather than thematic attributes. One control associated with line simplification is also the number of classes, although, in this case, each class is represented by a parameter that is the line segment connecting the lowest- and highest-ordered endpoints (it is assumed that the coordinates comprising the line have been ordered in a list from 1 to N) of any line segments in the group, rather than a single data value such as the mean or median. The fewer the number of classes, the more generalized is the resultant simplification. More frequently, however, other controls such as a bandwidth tolerance or a desired level of some geometric property of the line are used.

Hierarchical simplification again imposes the further control that members of the same group in an N-group simplification must be members of the same group in an (N-1)-group simplification. Hierarchical methods have been used to encode lines as strip trees (Ballard 1981). Jones and Abraham (1986, 1987) have also proposed using the strip-tree approach in the design of line-simplification procedures for a scale-independent data base. In the strip-tree approach, it is assumed that the top of the tree is associated with the most generalized line, i.e., a line segment connecting the two endpoints of the line. In a top-down framework, such as a strip tree, each point enters the tree as a node that splits the first line segment into two new segments, based on some geometric property. Because the strip-tree approach uses a bandwidth objective to simplify the line, the point having the maximum displacement from the line segment connecting the endpoints of the line segment would be added as the next node to the simplification tree.

The following example illustrates this process. An encoded string is given in Figure 3 that has 27 points. The top of a bandwidth simplification tree for this string is the line segment (1-27) connecting points 1 and 27. At each step of building this tree, the new node will correspond to the point that has the greatest vector displacement to the current line segment. Initially, point 16 has the greatest displayed placement from the initial line segment and is used to partition the initial line segment into two segments (1-16) and (16-27); its displacement value, 180.27, is added to the tree as an attribute tag. Next, point 5 has the maximum displacement to the segment (1-16) and thus partitions this segment on the next iteration into segments (1-5) and (5-16); its displacement value of 100.82 is added to the tree as the attribute tag for its node. Point 21 has the maximum displacement to the other segment (16-27) and partitions it into two segments, (16-21) and (21-27); its attribute tag is 63.45. Figure 4 illustrates the successively lower levels of generalization as the number of retained

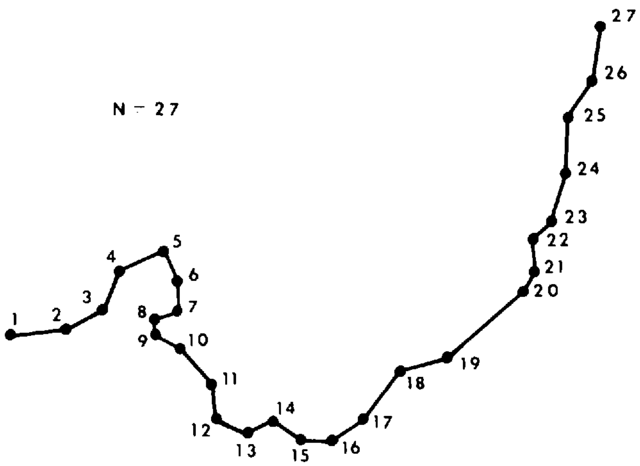


Figure 3. A sample encoded string.

points increases from 4 to 12; Figure 5 presents the entire simplification tree for the bandwidth criterion.

It should be noted that while strip trees are binary trees, it is more efficient for the purpose of simplification to use the point that splits each line segment as the new node of the tree. In this manner, the geometric attribute for each simplification level can be assigned directly to each point in the original encoding. This leads to more efficient retrieval operations. However, the resulting tree is not binary; whenever the point that splits a segment either immediately precedes or follows the point corresponding to its parent node in the order of the original line, then the tree will not bifurcate. This is illustrated in Figure 5, where point 2 follows point 1 and thus does not bifurcate, and again, when point 4 precedes point 5, no bifurcation occurs.

In general, hierarchical line simplifications can proceed from either a top-down or a bottom-up approach. In addition to the bandwidth objective, other geometric properties of the line can be used to build the tree. If the objective were to maximize line length, the interior point associated with the two line segments having the maximum line length would be added as a node to this simplification tree. Next, each of these line segments is subdivided into two new segments by adding the respective interior point that contributes the most to the desired geometric property of the simplification.

If the objective is to maximize length of the simplified line, then the tree may be different from one built using a bandwidth criterion. In the above example, however, the structure of the line-length tree only differs from the bandwidth objective by reversing the order of point nodes 10 and 11, so that point 11 is above point 10. A different simplification tree exists for each geometric criterion used to caricaturize the original encoded string. Although the tree structure may be almost the same, the order in which

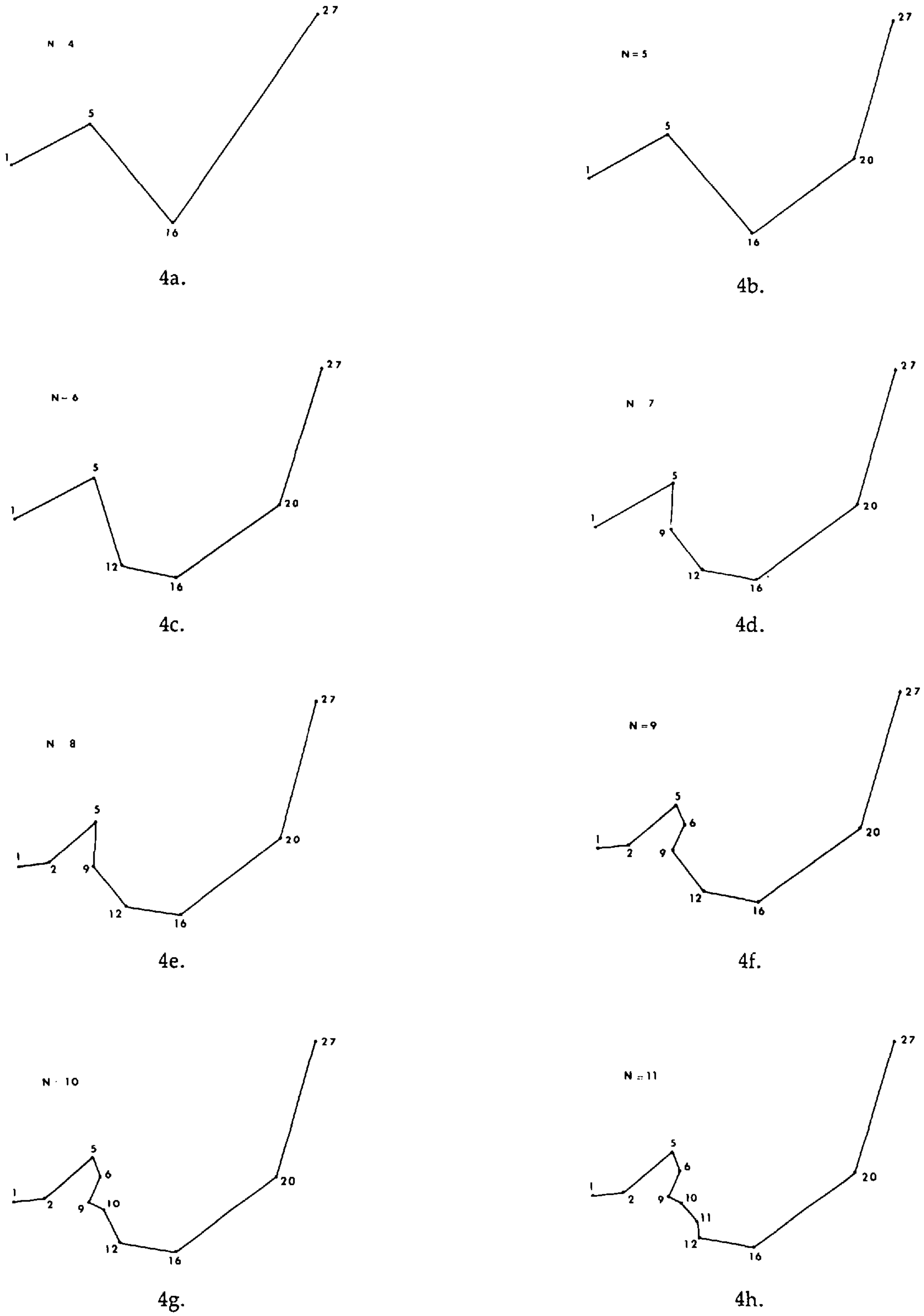
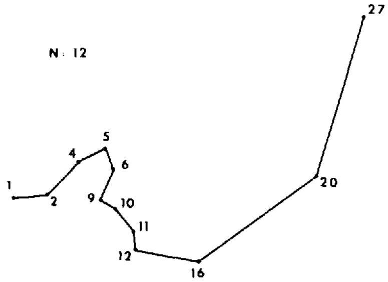


Figure 4. Successive levels of simplification.

points are retrieved may differ based on their attribute tag values.

Once the simplification tree has been constructed, then retrieving the points satisfying any desired level of generalization becomes a node-fathoming problem. For any desired geometric attribute level, the tree is searched along each branch until a node is fathomed. A node is fathomed when no further searches in that direction will result in a feasible solution to the current level of generalization. For ex-



4i.

Figure 4. Successive levels of simplification (*continued*).

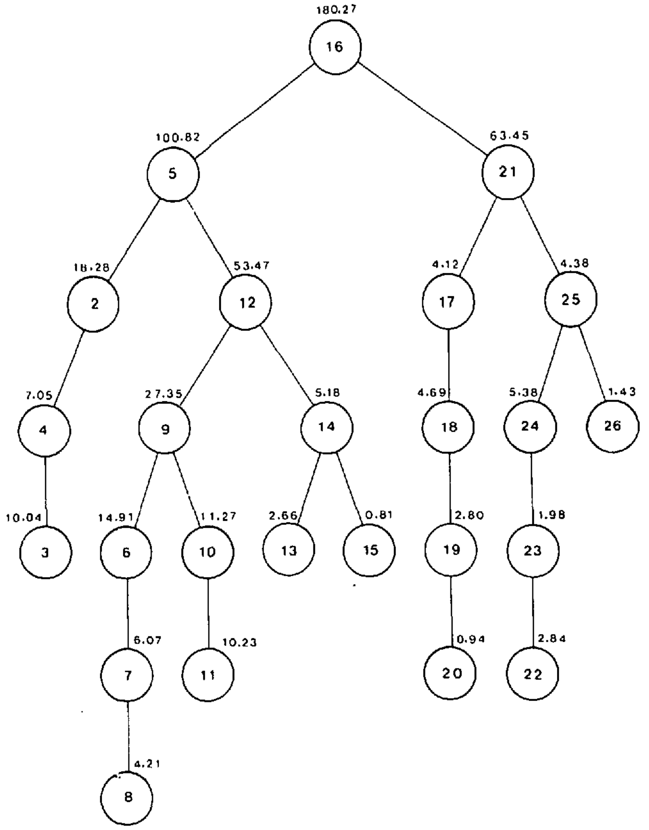


Figure 5. A top-down hierarchical tree based on the bandwidth criterion.

ample, suppose that using the bandwidth simplification tree given in Figure 5, it was desired to retrieve the string of points for which the deleted points were within 17 distance units of the resulting string. Starting with the first node of the tree, the attribute tag for that node is checked to see if it is less than the desired tag value. If it is not, the point associated with the node is included in the string and processing continues to the next successor node (when there is more than one successor node, the left one is examined first). If it is, then that node is redundant and that branch of the tree is fathomed. Processing backtracks to the next possible branch in the tree, and the search continues until all branches have been fathomed.

The attribute tag for node 16 (see Figure 5) is greater than 17, so in the first step, point 16 is retrieved and added to the string comprising the simplified line (see Figure 6). Next, node 5, the left successor to node 16, is examined; its attribute value is also greater than 17, so point 5 is added to the string in this step and its successor node is then checked. The attribute value for node 2 also passes the test and point 2 is added to the string in the third step (Figure 6). The successor to node 2, node 4, has a value less than 17, so this branch is now fathomed. Because node 2 has no other successors, control backtracks up the tree to node 5 and then to its right successor, node 12. This process continues until all the branches are fathomed, and a total of six interior points are added to the two endpoints to form the simplified string (Figure 6).

This technique works to ensure that the retrieved string will have at most the desired level of a geometric property, such as areal displacement or total vector displacement. It must be slightly modified if at least a desired level of a geometric property, such as line length or angularity, is wanted. In this case, nodes are retrieved along each branch of the tree until the corresponding attribute tag is greater than the desired amount; backtracking occurs to the next possible branch. At the end, one last node is then added to the string that corresponds to the extra segment needed to ensure that the final geometric value for the string is just greater than the desired level.

Even further computational efficiencies are possible by retrieving points in a list structure. Because each interior point in an original encoding has an associated geometric attribute tag, a linear check can be made of each point in sequence, and those points whose tag satisfies the criterion will be retrieved. For example, points 2 to 26 and their respective attributes for the bandwidth tree are given in the list in Table 1a. If the criterion is again to retrieve a simplified line satisfying a 17-unit bandwidth, all points having attributes greater than 17 will be retrieved. In Table 1b, points 2, 5, 9, 12, 16, and 21 meet this objective and are retrieved, as well as the endpoints 1 and 27. If the geometric attribute monotonically decreases from

Initial segment ——→ 1 27

First step ——→ 1 16 27

Second step ——→ 1 5 16 27

Third step ——→ 1 2 5 16 27

Fourth step ——→ 1 2 5 12 16 27

Fifth step ——→ 1 2 5 9 12 16 27

Sixth step → 1 2 5 9 12 16 21 27

Figure 6. An example of the retrieval process.

Table 1a. A list of points and their associated bandwidth value.

Point ID	Value	Point ID	Value	Point ID	Value
2	18.28	11	10.23	19	2.80
3	10.04	12	53.47	20	0.94
4	7.05	13	2.66	21	63.45
5	100.82	14	5.18	22	2.84
6	14.91	15	0.81	23	1.98
7	6.07	16	180.27	24	5.38
8	4.21	17	4.12	25	4.38
9	27.35	18	4.69	26	1.43
10	11.27				

Table 1b. The retrieved list of points with bandwidth values greater than 17.

Point ID	Value
2	18.28
5	100.82
9	27.35
12	53.47
16	180.27
21	63.45

one level to the next level down the tree for minimizing objectives or increases for maximizing objectives, then the same number of coordinates will be retrieved in this method as in the tree-fathoming one. This property is not present for the bandwidth objective. In Figure 5, node 3 has an attribute value of 10.04, while its predecessor, node 4, has an attribute value of 7.05. Node 3 would be added to the simplifying string in this method, although its branch would have been fathomed at node 4 in the other procedure. Thus, a few extra points may be retrieved than in the fathoming method. However, line-length trees have the desired property because of the triangle inequality rule—two sides of a triangle are always greater than the third side. Thus, no more points would be retained than in the fathoming method.

In addition, the points can be sorted by the geometric attribute. In a sorted list, either a predetermined number of points can be immediately retrieved or only a portion of sorted points is checked until the geometric value for a particular point is less than or greater than the desired attribute level. For example, the sorted list for Figure 5 is given in Table 2a. If it were desired to retrieve only eight interior points for a given scale representation, then the first eight points of this list would be retrieved. Alternatively, if the bandwidth tolerance was set at a value of 11, the first eight points also would have been retrieved (Table 2b). The sorted list reduces the need to search the entire list of points; this can improve computation if there are an extremely large number of points in the original list and if a higher level of generalization is desired. Besides the computational efficiencies gained by this approach, storage savings would also accrue because the tree itself would not have to be stored, just the tree attributes.

Conclusions

An important preprocessing operator in a digital mapping system is line simplification. Although many different simplification algorithms have been proposed to accomplish this task, each solution is unique to a given scale, specified at the outset. This paper has presented the advantages of general hierarchical approaches to line simplification. Hierarchical classification methods are frequently used to group objects by their thematic attributes. A similar method is now used for the other generalization problem of line simplification. The advantage of this approach is that a set of line caricatures corresponding to all tolerance levels is calculated in a single stage and compactly stored in a tree structure. The simplified line for any desired tolerance is easily retrieved by traversing the tree. Additional computational and storage savings are also possible by storing the tree attributes in a list or sorted structure and retrieving the desired coordinates in a linear search.

REFERENCES

Ballard, D. 1981. "Strip Trees: A Hierarchical Representation for Curves." *Communications,* vol. 24, pp. 310-321. American Association for Computing Machinery.

Burrough, P. 1986. *Principles of Geographical Information Systems for Land Resources Assessment.* Oxford, England: Clarendon Press.

Cromley, R., and G. Campbell. 1991. "Noninferior Bandwidth Line Simplification: Algorithm and Structural Analysis." *Geographical Analysis,* forthcoming.

Cromley, R. 1988. "A Vertex Substitution Approach to Numerical Line Simplification." *Proceedings,* Third International Symposium on Spatial Data Handling, pp. 57-63.

Douglas, D., and T. Peucker. 1973. "Algorithms for the

Table 2a. The sorted list of points and their associated bandwidth value.

Point ID	Value	Point ID	Value	Point ID	Value
16	180.27	11	10.23	8	4.21
5	100.82	3	10.04	17	4.12
21	63.45	4	7.05	22	2.84
12	53.47	7	6.07	19	2.80
9	27.35	24	5.38	13	2.66
2	18.28	14	5.18	23	1.98
6	14.91	18	4.69	26	1.43
10	11.27	25	4.38	20	0.94
				15	0.81

Table 2b. The retrieved list of first eight points (bandwidth values greater than 11).

Point ID	Value
16	180.27
5	100.82
21	63.45
12	53.47
9	27.35
2	18.28
6	14.91
10	11.27

Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature." *The Canadian Cartographer*, vol. 10, pp. 112-122.

Freeman, H. 1978. "Shape Description via the Use of Critical Points." *Pattern Recognition*, vol. 10, pp. 159-166.

Jenks, G., and F. Caspall. 1971. "Error on Choroplethic Maps: Definition, Measurement, Reduction." *Annals*, vol. 61, pp. 217-244. Association of American Geographers.

Johnston, R.J. 1976. *Classification in Geography*, CATMOG Series No. 6. Norwich, England: Geo Abstracts Ltd.

Jones, C., and I. Abraham. 1986. "Design Considerations for a Scale-Independent Cartographic Database." *Proceedings*, Second International Symposium on Spatial Data Handling, pp. 384-398.

Jones, C., and I. Abraham. 1987. "Line Generalisation in a Global Cartographic Database." *Cartographica*, vol. 24, no. 3, pp. 32-45.

McMaster, R. 1983. "A Mathematical Evaluation of Simplification Algorithms." *Proceedings*, Auto Carto VI, vol. 2, pp. 267-276.

McMaster, R. 1986. "A Statistical Analysis of Mathematical Measures for Linear Simplification." *The American Cartographer*, vol. 13, pp. 103-117.

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

McMaster, R. 1987b. "The Geometric Properties of Numerical Generalization." *Geographical Analysis*, vol. 19, pp. 330-346.

Peucker, T. 1976. "A Theory of the Cartographic Line." *International Yearbook of Cartography*, vol. 16, pp. 134-143.

White, E. 1985. "Assessment of Line-Generalization Algorithms Using Characteristic Points." *The American Cartographer*, vol. 12, pp. 17-27.

Zycor Inc. 1984. *Manual and Automated Feature Displacement*, Report for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia.