A Progressive Line Simplification Algorithm*

Qing Sheng Guo
Christoph Brandenberger
Lorenz Hurni

Institute of Cartography
ETH Honggerberg, ETH Zurich
CH-8093, Switzerland
+41-1-6333033 (phone)
+41-1-6331153 (fax)
Qingshengguo@yahoo.com(Department of Cartography, Wuhan University, 430079)
[Guo, Brandenbergen, and Hurni]@karto.baug.ethz.ch

Abstract

Line simplification is an important topic in the process of map generalisation covered by numerous research groups. In this paper, the authors will report about an application of line simplification considering spatial knowledge. A method for identifying potential conflict regions, in order to avoid the self-intersection of generalized lines, is also discussed. Furthermore a new progressive line simplification algorithm is presented. From the view point of spatial cognition, a spatial hierarchical structure is proposed, and its application for construction of spatial knowledge in relation to a line is explained.

Keywords: Progressive Line Simplification, Map Generalisation, Spatial Knowledge, Spatial Cognitive, Spatial Hierarchical Structure, Potential Conflict Region, Self-intersection of a Line.

1. Introduction

Map generalization is one of the classical cartographic problems. All maps, whether in digital or analogue form, are generalized representations of the reality. Generalisation is necessary in order to improve the display quality of small scaled maps; to allow analysis with different grades of detail; and to reduce data storage requirements (Joao, 1998). Line simplification algorithms can be divided into five categories: independent point algorithms, local processing algorithms, constrained extended local processing algorithms, unconstrained extended local processing algorithms and global algorithms (Kreveld/Nievergelt/Roos/Widmayer, 1997). In this book, numerous line generalisation algorithms in use are also discussed.

Using maps the geographic spatial knowlege can be communicated. The result of map generalization should always retain the important geographic spatial knowledge, according to the conditions of the new small scaled map (McMaster/Shea, 1992). Therefore, the existing spatial knowledge related to the line should be maintained in the process of line generalisation. In many researches the different existing line generalisation algorithms are compared with each other (Visvalingam/Herbert, 1999; Dutton, 1999(a)). Because spatial line knowledge is essential in generalisation, researchers have investigated the classification and the application of cartographic knowledge for line generalisation (Dutton, 1999(b); Kilpeläinen, 2000; Weibel, 1996; Lechthaler/Kasyk, 1999). Furthermore the self-intersection of a generalized line is another problem in line generalisation. In order to overcome it, some algorithms have been evaluated (Li/Sui, 2000; Dutton, 1999(a); Christensen, 1999/2000; Saalfeld, 1999).

In this paper, the geometric knowledge related to cartographic lines is discussed. By means of computaional geometry algorithms, the global and local characteristics of cartographic line features, as for example, the convex line hull, can be gained. At the moment, the integration of these characteristics

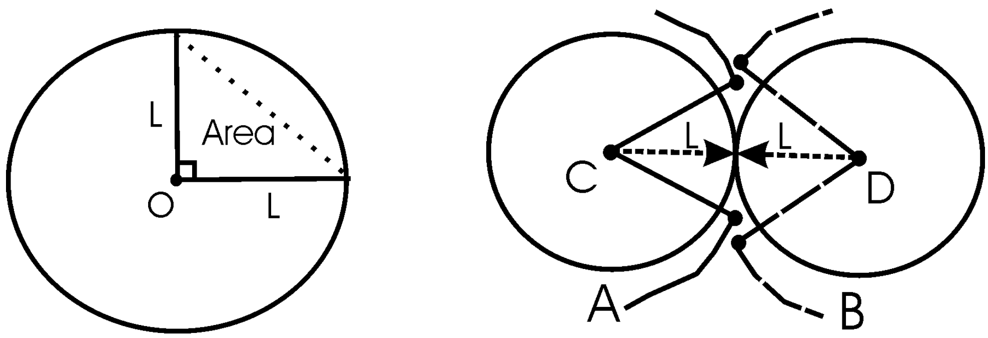
* This research is supported by NSF, No. 49971068, by National Excellent Young Teacher Fund, by Huo Yingdong Fund for Young Teacher in University, by the Swiss Confederation and by the Institute of Cartography at ETH Zurich.

into the process of line generalisation is one of the main research topics in digital cartography. The spatial characteristic line points should be ordered, according to their importance, and composed to a spatial hierarchical structure. To overcome the problem of line self-intersection within the process of generalisation, an identification routine for them is introduced.

2. Identification of potential conflict regions

Through line simplification, often self-intersections of the generalized lines occur, because the elimination of vertices influence the adjacent vertices and the spatial relation between them. Thus the potential conflict region with generalized self-intersecting polylines should be identified, and tested for self-intersection. In fig. 2, different local situations of potential conflict regions are shown. All vertices O in fig. 2 are points which will be tested. Adjacent vertices laying inside a circle were searched, and the direction of this line is determined (fig. 2). The algorithm for identification of potential conflict regions runs as follows:

(1) According to a given triangle area as a threshold, the influence distance of a vertex is computed in form of a circle. In our test, this influence distance equals to 2*SQRT(2*Area), with Area as the given minimal triangle area. Because the influence scope of an eliminated vertex is SQRT(2*Area) (Fig. 1 (a)), and the influence distance of two vertices is the double influence scope of a vertex (Fig.1 (b)). The start point of an original cartographic line is the first point, that will be tested, for example, point O in fig. 2.



Area is a given triangle area as a threshold.
L is the influence distance
L=SQRT(2*Area)

C and D will be eliminated
L is the influence distance of a vertex
Line sections A and B belong to a line

(a) (b)

Fig. 1 The influence distance of a vertex

(2) The vertices, that are laying inside or outside the circle, are identified (fig. 2(a)). The vertices, for example, vertices A, B and C (fig. 2(a)), are very important. These vertices are used to identify vertices influenced by the vertex O. But vertices E and F in fig. 2(a) will not be detected, because vertices between vertex E and F will be tested in the next step, these vertices must be in the potential conflict region.

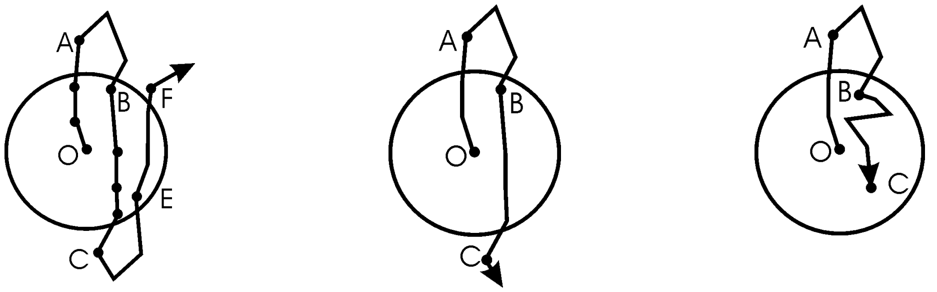
(3) If a vertex C exists (fig. 2(b)), and vertex C is not the end point of the original line, then all vertices inside the circle are regarded as a potential conflict partners of vertex O (fig. 2(b). Then the existing potential conflicts between vertex O and these partners are stored, as basis for solving the self-intersection problem of generalized lines.

(4) In fig. 2(c), vertex C is inside this circle, and the end point of this original line. If the length of the line section BC (fig. 2(c) is greater than a given threshold, then all vertices within the circle belong to the potential conflict region of vertex O (fig. 2(c). In this test, this pregiven threshold is two times the circle radius. By a closed original cartographic line, the length of line section inside this circle and the distance between the two end points of this line section are calculated. If this length is greater than twice the

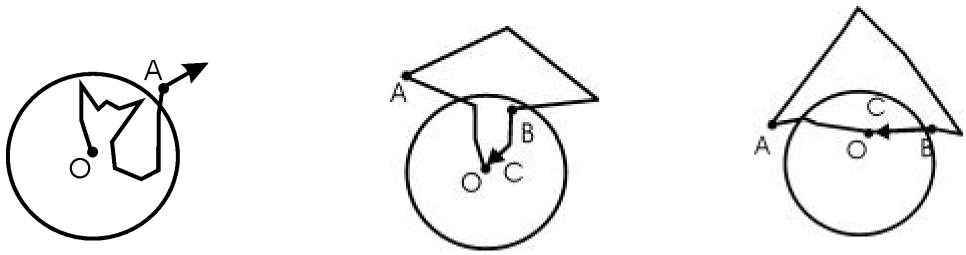
radius, or this distance is greater than the radius, then the potential conflict region can be identified, for example fig. 2(e) shows a potential conflict region, but fig. 2(f) does not.

(5) If only vertex A exists(fig. 2(d)), and the length of the line section inside this circle is greater than a certain given length, for example, three times of the radius in our test, then a potential conflict region of the vertex O can be identified.

(6) The adjacent sequent vertices of vertex O will be tested. If all vertices to be tested are inside this circle, then this process ends. Fig. 3 (b) shows the points in a potential conflict region of a line. Fig. 3 (a) shows samples of circles without potential conflict regions.



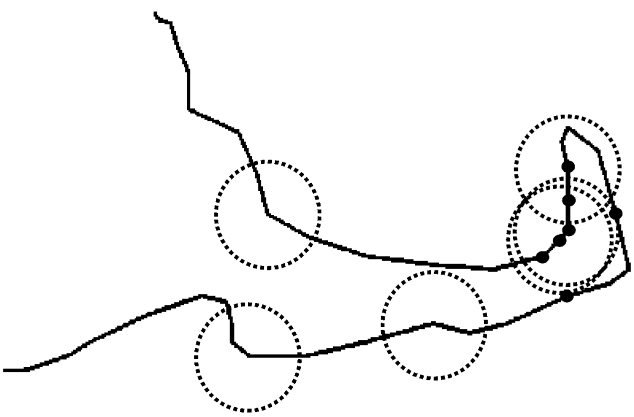
(a) (b) (c)



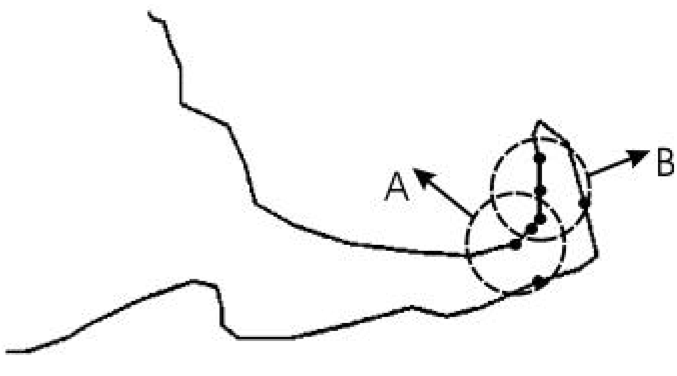
(d) (e) (f)

The radius of the circle is the influence distance

Fig. 2 Identification of potential conflict regions



(a)
Samples of circles without potential conflict regions



(b)
A and B are potential conflict regions

Fig. 3 Points situated inside a potential conflict region

3. Spatial knowledge and spatial cognition in line simplification

In a generalized map, the geographic spatial knowledge should be correctly communicated. Therefore, the geographic spatial knowledge of the map must be retained in the process of map generalization. In the test carried out, characteristic points of a line and their spatial cognition are investigated.

3.1 Maximal line points

There exist two kinds of characteristic line points, the first which consists of local maximal points and a second with points having a local direction change in X- or Y- direction. In order to find out local maximal points from a line, the inflection points of the line must be firstly identified. Moreover it is necessary to integrate new inflection points during the process of inflection point identification, for example point A in fig. 4(b). Otherwise, the local maximal point is also at the same time the inflection point. In fig. 4(a) point A is a local maximal point and also an inflection point, because an inflection point is a vertex with a direction change between negative and positive curvature in a line. In fig. 4(b) point A is a inflection, that is a new inserted point, and point B is a local maximal point. After the search of the inflection points, the local maximal points can be determined. Fig. 4 (c) and (d) show the local maximal points of two lines. A local maximal line point is a point, that has the largest perpendicular distance to the base line segment, whose end points are two adjacent inflection points, than other original vertices between two neighbouring inflection points. In fig. 4(e), this principle is shown, and vertices A and B are inflection points. Fig.5 shows the local maximal points and the end points of a line.

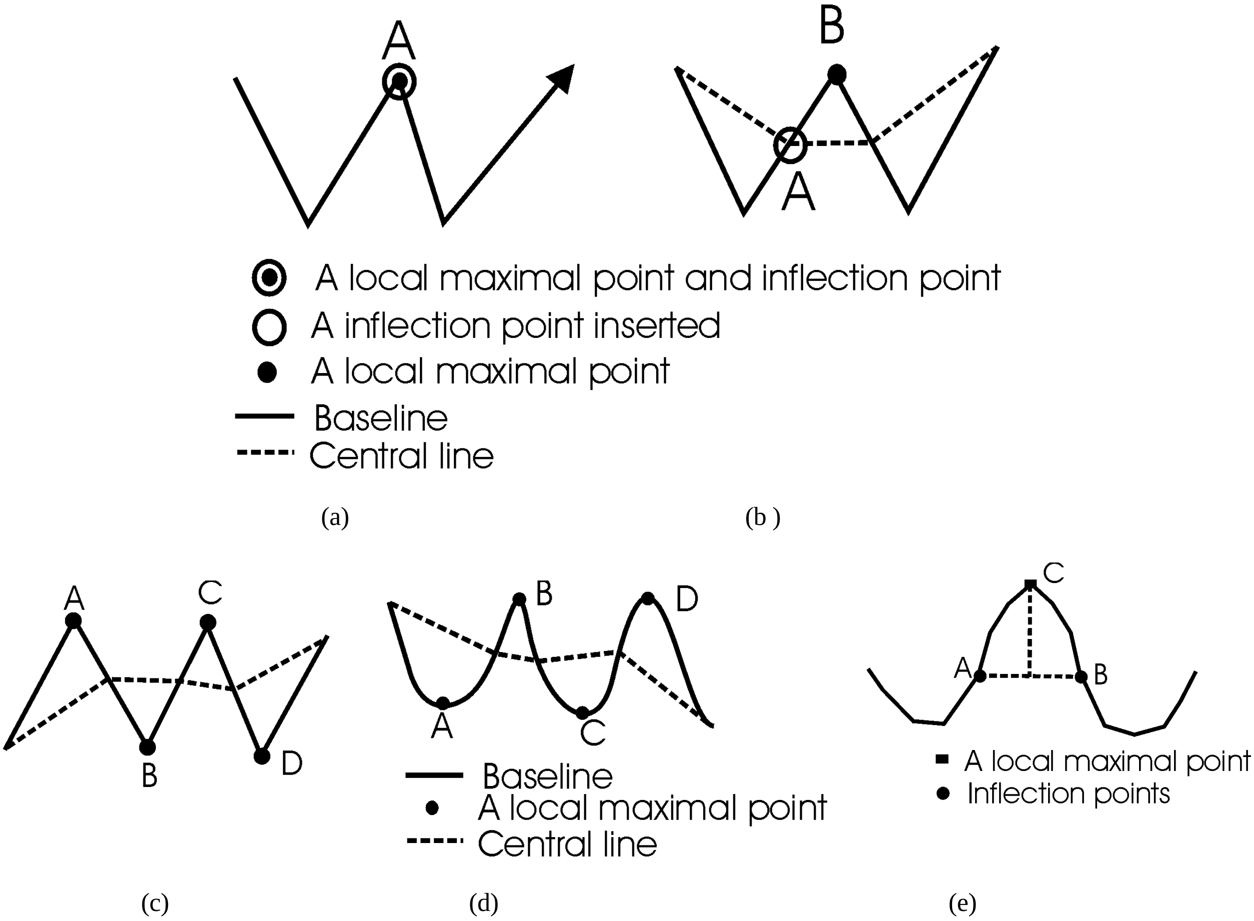
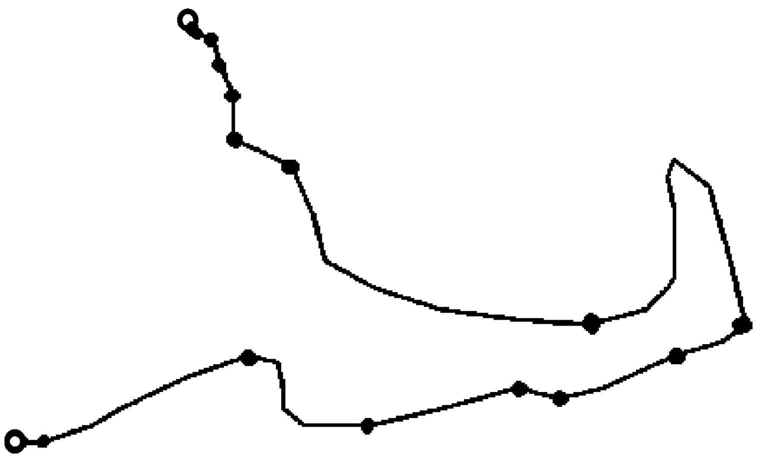
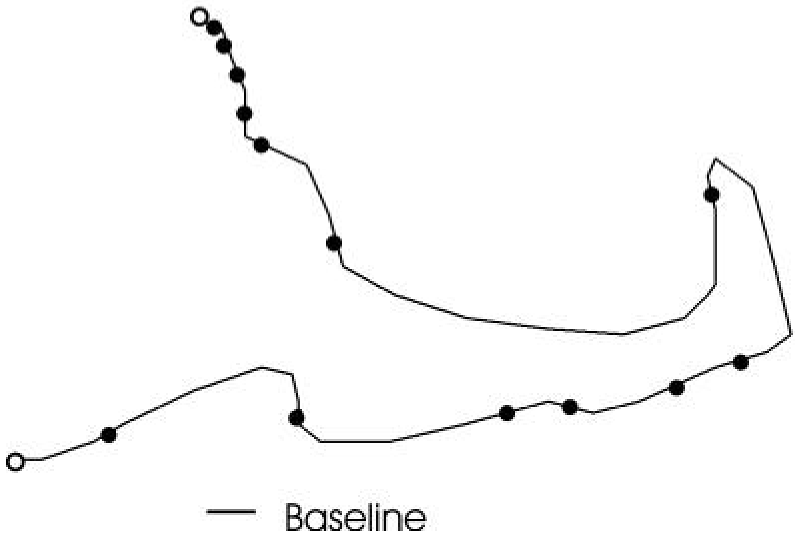


Fig. 4 Inflections and local maximal points



(a) Local maximal points and the line end points



(b) Inflection points and line end points
Fig. 5

A point with a local direction change in X- or Y- direction can be identified by means of the comparision of X (or Y) coordinates of a vertex with the coordinates of its adjacent vertices. Fig. 6 shows the basic principle for identifying vertices with a local direction change. In fig. 6(a), the X coordinate of the considered vertex is less or equal than the X coordinates of all adjacent vertices, or is greater or equal than the X coordinates of all adjacent vertices. In fig. 6(b), the Y coordinates are applied for this comparision. Fig. 7 shows all points of a line, having a local direction change.

(a) (b)

Fig. 6 Identification of local direction change

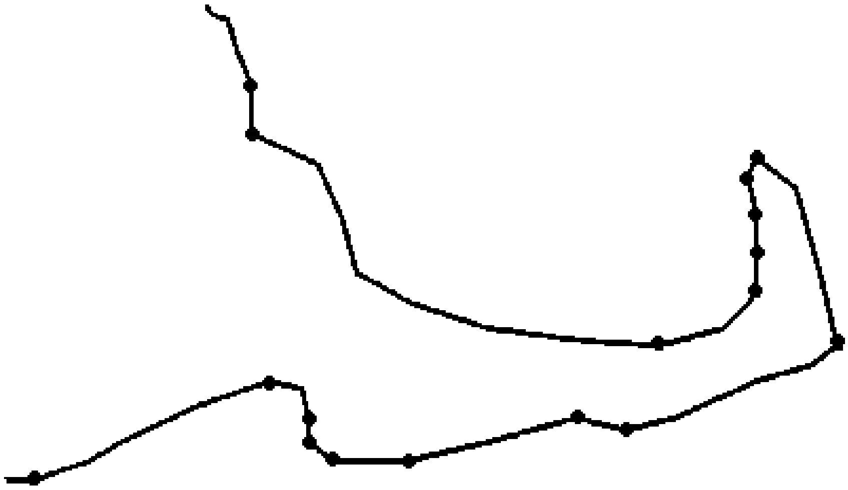


Fig. 7 Points with a direction change

3.2 The characteristic points of a convex hull

In computational geometry text books, many algorithms for building the convex hull are given. In this paper, the algorithm by Laszlo (1996) is applied on the test data set. Before a line could be generalized, points from its convex hull must be extracted and stored for further treatment. In fig. 8 the convex hull of a line is represented.

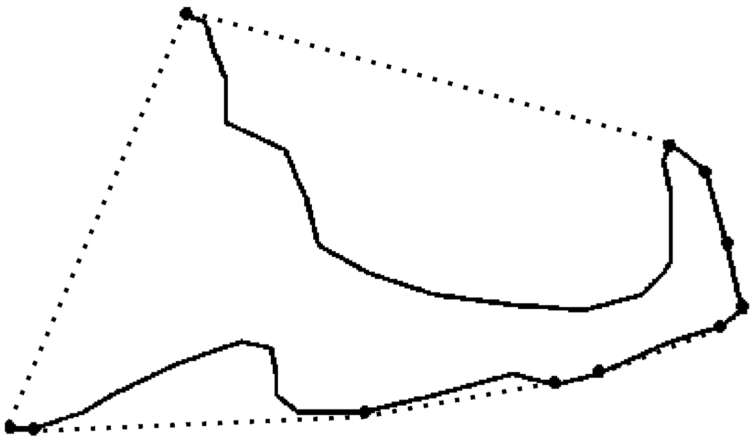


Fig. 8 The convex hull of a line with its characteristic points

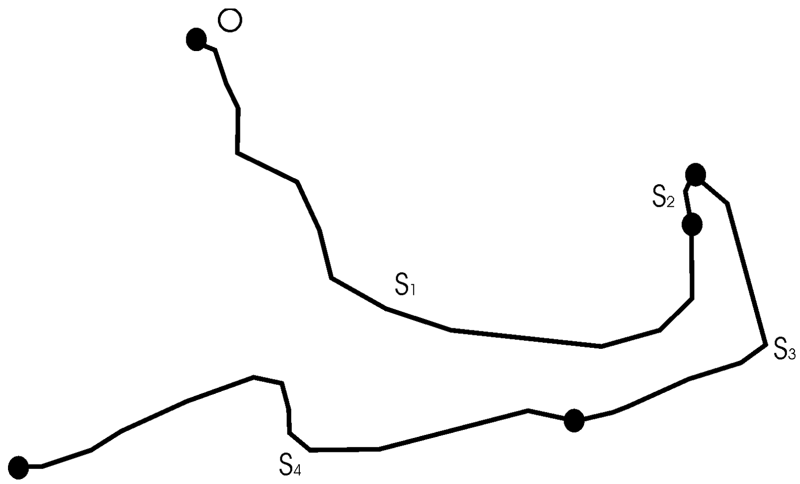
3.3 The line segmentation based on monotone lines

There are many identification algorithms for monotone lines. A monotone line is defined as a line, with progressively increasing Y- (or X-) coordinates. The algorithm for monotone line segmentation runs as follows:

(1) The start point of an original cartographic line is selected, and linked to the following sequent line vertices. Then, the corresponding monotone polylines in X- and Y-axis direction could be evaluated.

(2) These two identified monotone polylines are compared with each other, and the end point of the longest polyline will be stored.

(3) This stored end point is used as the start point of the next monotone polyline. This process is repeated until the end point of the actual monotone polyline is identical with the end point of the original line. In fig. 9 the result of a line segmentation shown.



O is the start point, S_1, S_2, S_3 and S_4 are X- or Y- monotone polylines

Fig. 9 The line segmentation based on monotone lines

3.4 Spatial cognition of characteristic points

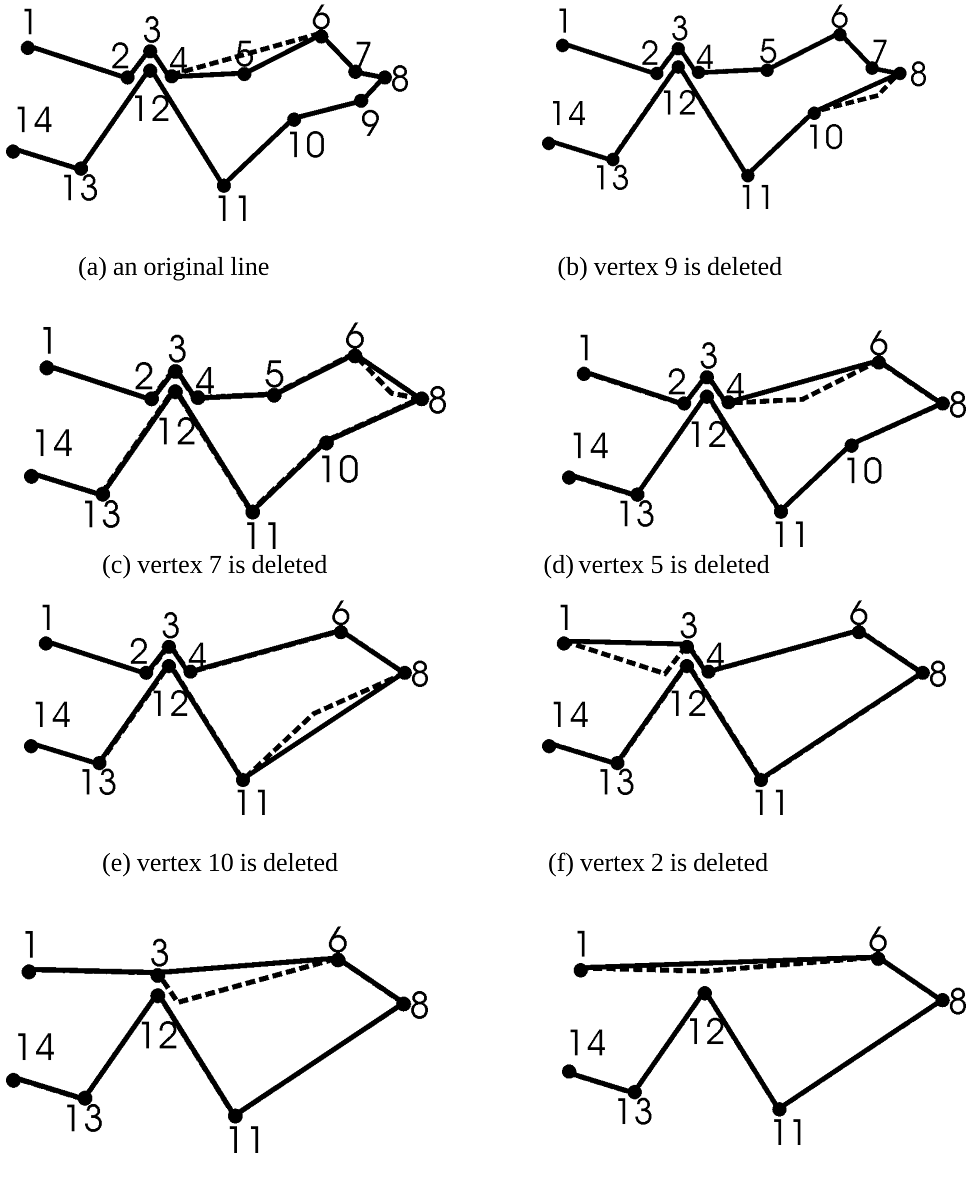
The spatial characteristic line points have different degrees of importance for the process of line simplification. Some points express local line features, for example, local maximal points, others express global line features, for example, points on a convex hull. Therefore, the spatial characteristic line points build a hierarchical structure. In this test, four different kinds of characteristic line points are considered. These are: the local maximal points, the points with direction change, the points between two monotone polylines and the points of the convex hull. The importance of vertices is assigned to three categories. The classification for all points algorithm runs as follows:
(1) Firstly, the degree of vertices importance is set to zero.
(2) If a vertex is a local maximal point or a point with a direction change, then the vertex degree is altered to one.
(3) But if a vertex is a local maximal point and also a point with a direction change, then the degree of the vertex is changed to two.
(4) If a vertex is a point between two adjacent monotone polylines, and its degree is equal to zero, then the vertex degree is also set to two. If its degree is not equal to zero, then the vertex degree is set to three.
(5) For points of the convex hull, the degree is three. The degree of end points and other fixed points of the line is four.

4. The basic principle of progressive line simplification

In this section, a progressive line generalisation algorithm will be discussed, that is similar to them of Visvalingam and Whyatt (Visvalingam/Whyatt, 1993). This algorithm should be explained by means of a very simple test data set (fig. 10(a)), and only one important vertex is fixed, for example, vertex 8 in fig. 10(b). This algorithm runs as follows:

(1) Every vertex begining with the fixed vertex 2 to vertex 13 and its two adjacent vertices form a triangle (fig. 10(a)). For example, vertices 4, 5 and 6 in fig. 10 (a) build a triangle. The size of this triangle corresponds to the degree of the vertex 5. The area of this triangle can be determined. For simplifying the algorithm explanation, only vertex 8 in fig. 10(b) is selected as a significant vertex vertex.

(2) When vertices are laying inside this triangle, for example vertex 12 in fig. 10(a), or for a vertex, which belongs to the categorie of important vertices, the area of the triangle can not be compare with areas of other triangles. In this case, this vertex might not be deleted.
(3) The triangle areas are compared with each other until the smallest is found. The corresponding vertex can be deleted, and a new line is formed. For example, after deleting vertex 9 in fig. 10(a), the new line in fig. 10(c) is defined.
(4) The new areas of the triangles, which belong to two adjacent vertices of the deleted vertex, for example, the two adjacent vertices 8 and 10 of the vertex 9 in fig, 10(c), must be recomputed.
(5) This process returns to step (2), until the smallest triangle area is greater than the given threshold. Then it will stop. The workflow of this process is visualized in fig. 10(c) to fig. 10(i).



(a) an original line (b) vertex 9 is deleted

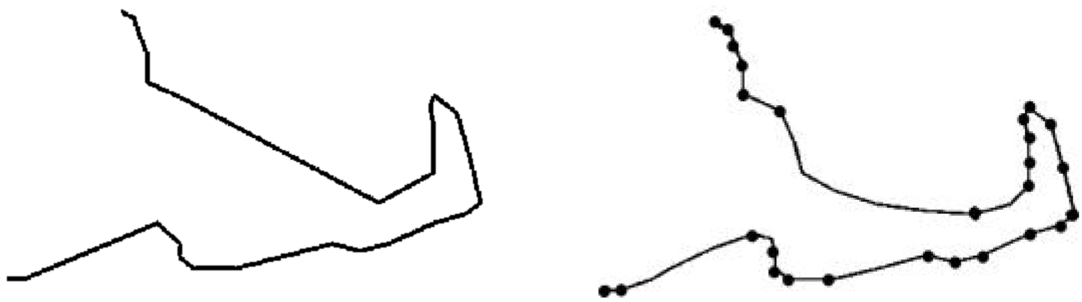
(c) vertex 7 is deleted (d) vertex 5 is deleted

(e) vertex 10 is deleted (f) vertex 2 is deleted

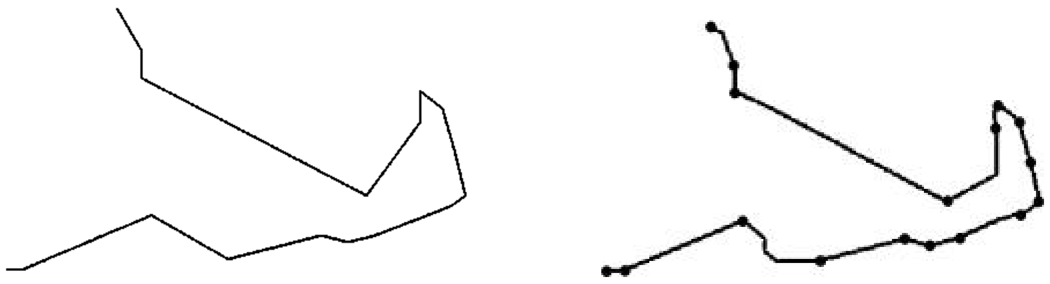
(g) vertex 4 is deleted (h) vertex 3 is deleted

Fig. 10 The basic principle of progressive line simplification

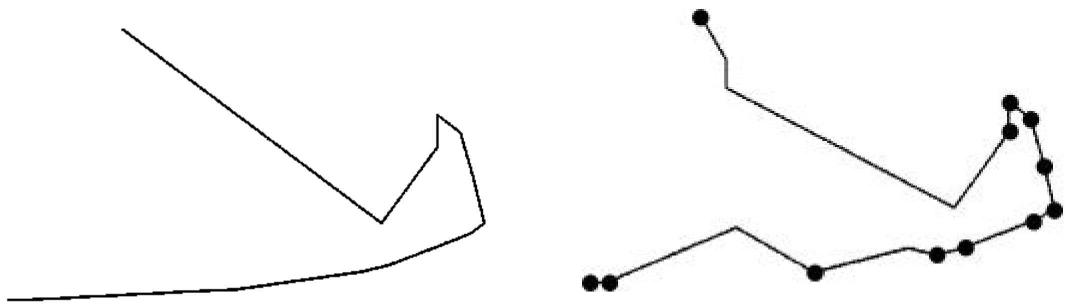
Fig. 11 shows the complete process of progressive line simplification. In fig. 8 the original test data are represented. For the line vertices the degrees of importance as mentioned above are evaluated. Fig. 11(a) shows the line sections between two characteristic points, with a degree of importance of one, after generalisation, and the new generated line (fig. 11(a)). In fig. 11(b) and (c), the line generalized sections between two characteristic points, with a degree of importance of two or three, are also generalized. Fig. 11(d) shows the result of the entire line generalisation.



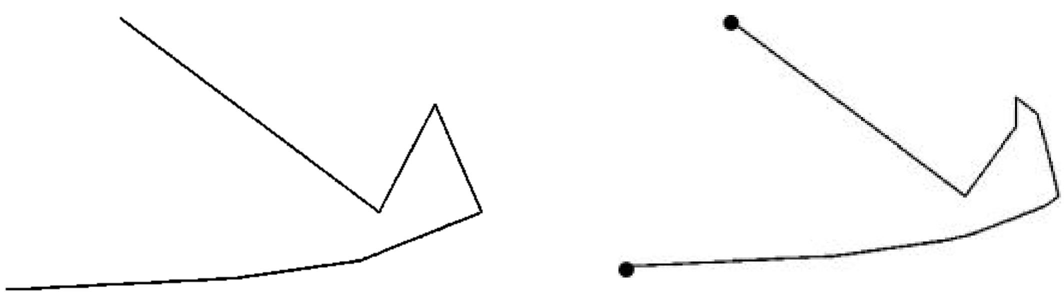
(a) All characteristic line points are considered (a1) Points with important grades>=1



(b) Only line points of grades>=2 are considered (b1) Points with important grades>=2



(c) Only line points of grades>=3 are considered (c1) Points with important grades>=3



(d) The generalized line (d1) The end points

Fig. 11 The various steps of the progressive line simplification process under consideration of different point grades

5. A test of progressive line simplification

The above mentioned progressive line simplification algorithm has been applied on a complex test data set. In fig. 12, the original lines are visualized, which are downloaded from http: // crusty.er.usgs.gov / coast / getcoasts.html. In fig. 13 results produced by means of a progressive line simplification algorithm with different generalisation degrees are displayed. This line generalisation process is equal to the process in fig. 11. In fig. 13, the results possess the main important line features, according to the requirement of the different generalisation degrees, without any line self-intersection.

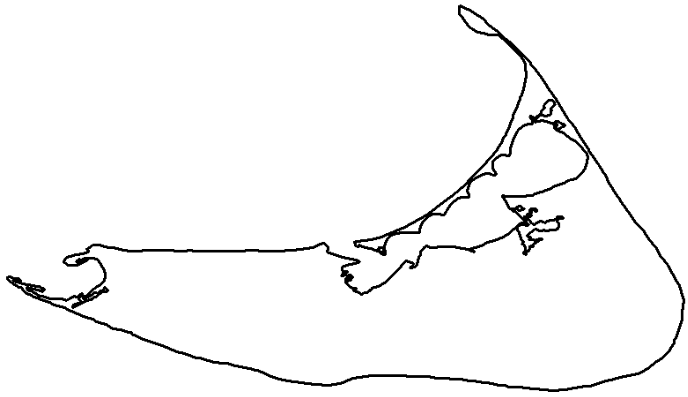


Fig. 12 The original test data

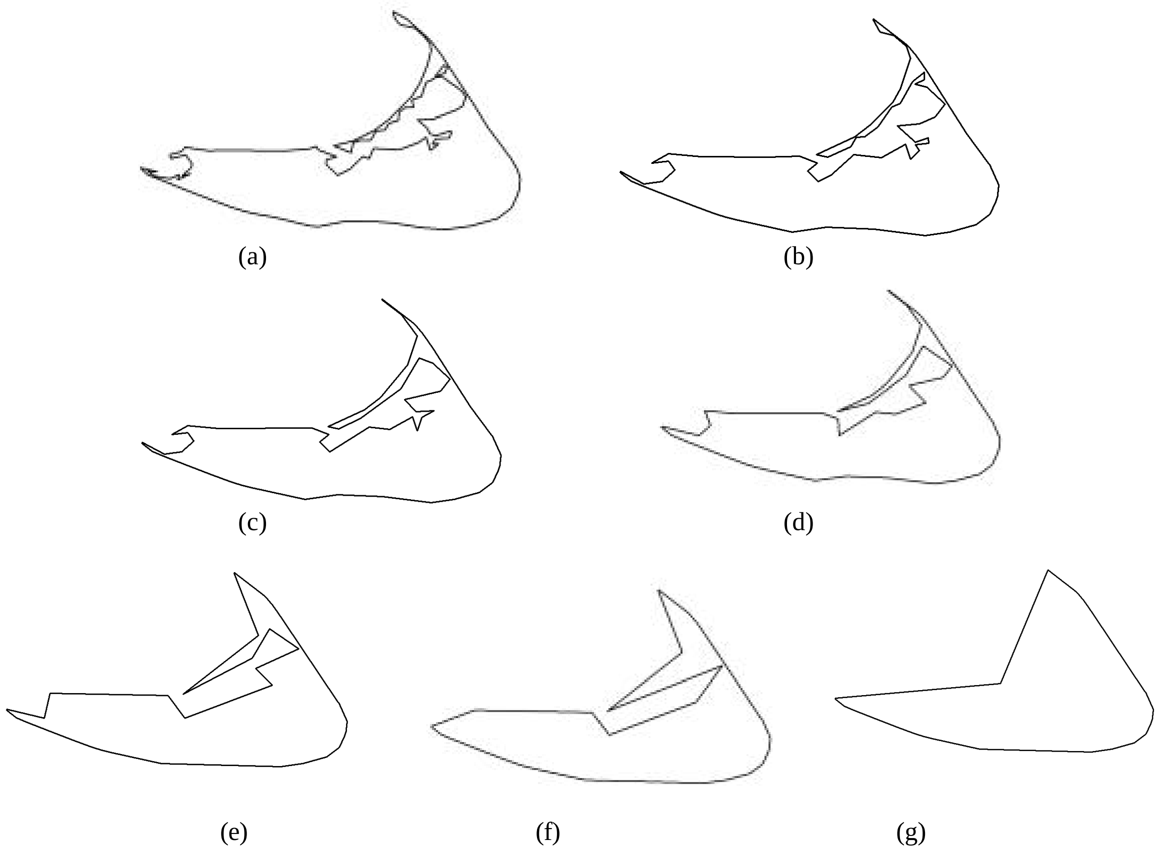


Fig. 13 Typical results of a generalisation with a progressive line simplification algorithm with considering line self-intersections

6. The self-intersection of generalized lines

The self-intersection of lines frequently occurs in line generalisation. In fig. 14, the result of a line generalisation by means of a Douglas-Peucker line simplification is represented. In fig. 11 the original test data is given. For a better program performance, mostly progressive line simplification algorithms do not consider the self-intersection problem in data handling. If self-intersections occur in the result, then the original corresponding line sections should be handled again with the progressive algorithm mentioned above in this paper.

To identify line segments which intersect each other is easy to realize. In fig.15, P_1, P_2, P_3, P_4 and P_5 are enumerated codes of vertices on an original line, and parts of an entire generalized lines with self-intersections are shown. In fig.15 (a), the line segment P_1P_2 intersects with the line segment P_3P_4. Therefore the original line section from point P_1 to point P_4 must be handled once more, with P_1, P_2, P_3 and P_4 as original line vertices, and P_2 > P1, $P_4 > P_3$. In case a line segment intersects with more than two other line segments, then the original line section is the polyline from the vertex with lowest number to the vertex with the heighest number in the related vertices, for example, an original line section from vertex P_1 to vertex P_5 (fig. 15(b)), and with $P_1<P_2<P_3<P_4<P_5$.

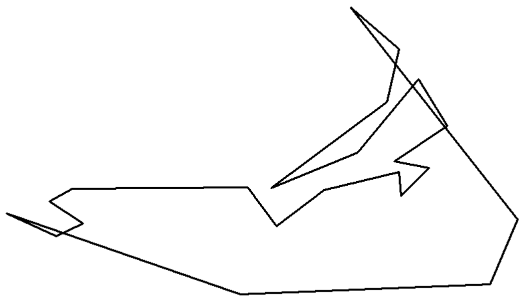
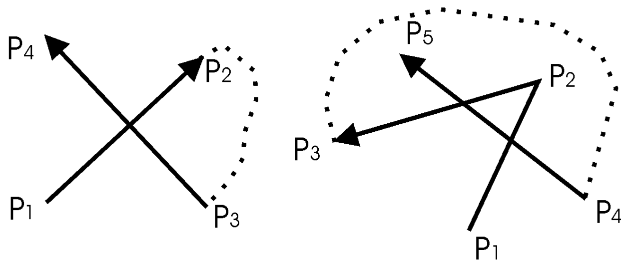


Fig.14 An typical example of a line with self-intersecting, mostly produced by an automatic generalisation without taking into account the problem of line self-intersection



(a) (b)

Fig. 15 The identification process of line self-intersections

7. Conclusion and discussion

From the presented results in this test, it can be concluded, that by considering the spatial knowledge the quality of line simplification could be essentially improved. This progressive line simplification algorithm considers the spatial knowledge related to a line. And the spatial characteristics of a line can be retained in the generalisation result according to the generalisation requirement. The self-intersections of a generalized line do not occur.

But there exist further spatial knowledge for lines, as example, spatial relations, which should be also considered. How to apply these in the process of line generalisation should be investigated in further research.

References

Christensen, A. H. J.: Line Generalisation by Waterlining and Medial Axis Transformation: Successes and Issues in an Implementation of Perkal's Proposal. The Cartographic Journal, Vol. 37, No. 1, June, 2000, pp. 19-28.

Christensen, A. H. J.: Cartographic Line Generalisation with Waterlinines and Medial-Axes. Cartography and Geographic Information Science, Vol. 26, No. 1, January, 1999, pp. 18-32.

Dutton, G. H.: A Hierarchical Coordinate System for Geoprocessing and Cartography. Springer-Verlag, Berlin, Heidelberg, 1999(a).

Dutton, G.: Scale, Sinuosity and Point Selection in Digital Line Generalisation. Cartography and Geographic Information Science, Vol. 26, No. 1, January, 1999(b), pp 33-53.

Joao, E. M.: Causes and Consequences of Map Generalisation. Taylor & Francis Ltd, 1998.

Laszlo, M. J.: Computational Geometry and Computer Graphics In C++. Prentice-Hall, Upper Saddle River, USA, 1996.

Li, Z. and Sui, H.: An Integrated Technique for Automated Generalisation of Contours Maps. The Cartographic Journal, Vol. 37, No. 1, June, 2000, pp. 29-37.

McMaster, R. B. and Shea, K. S.: Generalisation in Digital Cartography, Association of American Geographers, 1992.

Lechthaler, M. and Kasyk, S.: Theoretical and Practical Deliberations with Regard to Semi-Automated Cartographic Generalisation. The Cartographic Journal, Vol. 36, No. 1, June, 1999, pp. 59-65.

Saalfeld, A.: Topologically Consistent Line Simplification with the Douglas-Peuker Algorithm. Cartography and Geographic Information Science, Vol. 26, No. 1, January, 1999, pp. 7-18.

Van Kreveld, M., Nievergelt, J., Roos, T. and Widmayer, P. (Eds.): Algorithmic Foundation of Geographic Information Systems. Springer-Verlag, Berlin, Heidelberg , 1997, pp. 99-152.

Visvalingam, M. and Herbert, S.: A Computer Science Perspective on the Bendsimplification Algorithm. Cartography and Geographic Information Science, Vol. 26, No. 4, October, 1999, pp. 253-270.

Kilpeläinen, T.: Knowledge Acquisition for Generalisation Rules. Cartography and Geographic Information Science, Vol. 27, No. 1, January, 2000, pp. 41-50.

Visvalingam, M. and Whyatt, J. D.: Line Generalisation by Repeated Elimination of Points. Cartographic Journal, Vol. 30, No. 1, 1993, pp. 46-51.

Weibel, R.: A Typology of Constraints to Line Simplification. In: Kraak, M. J. and Molenaar, M. (eds): Advances in GIS Research II (Proceedings 7th International Symposium on Spatial Data Handling), London: Taylor & Francis: 9A.1-9A.14.