

A new approach and procedure for generalising vector-based maps of real-world features

Shaozhong Shi* and Martin Charlton

National Centre of Geocomputation, National University of Ireland, Maynooth, Ireland

(Received 14 April 2013; final version received 24 June 2013)

This paper presents a new approach and procedure for directly processing vector-based data sets to generalise maps depicting real-world phenomena. It shows how map generalisation and seamless matching of the edges of adjacent area features can be achieved. With this approach, the combined use of selected methods has led to a ground-breaking novel procedure for vector-based map generalisation guaranteeing outputting with neither self-intersection nor cross-intersection. The approach employs turning points and convex hull points as a set of characteristic points. These points define the shape and characteristics of real-world geographical features with delineating lines, which have complex rendering layouts, many turnings and inherent sub-features. The set of characteristic points are used as splitting points to partition lines into monotonic chains. Line simplification with an intuitive point reduction technique using the Douglas–Peucker algorithm is confined within monotonic chains to guarantee no self-intersection. In addition, the paper describes the most suitable and intuitive means to deal with gaps and overlaps potentially resulting from cross-intersections without adding or deleting any anchoring point. Distinctively different from the approaches of previous studies, the research was undertaken through an iterative process of exploratory programming, experimentation, validation and testing of results produced. This process was repeated until an integrated total solution was reached. It sheds new light on the fields of geocomputation, geographic information systems (GIS) and digital cartography.

Keywords: vector features; generalisation; line simplification; turning point; convex hull; monotonic chain; Douglas–Peucker

1. Introduction

Previous studies have shown that line simplification techniques such as the Douglas–Peucker algorithm can fail to maintain consistent or correct topological relations among vector map features. There are problems of self-intersections (Figure 1) and intersections when line simplification techniques are used on their own to automatically generalise maps. In recent years, there has been a revived interest in the application of the Douglas–Peucker algorithm and finding appropriate approaches and methods for topologically consistent map generalisation (e.g. Da Silva and Wu 2005; Corcoran, Mooney, and Winstanley 2011). A number of approaches and techniques for achieving topologically consistent map generalisation have been explored (Mantler and Snoeyink 2000; Saalfeld 1999; Wu and Marquez 2003; Zhang et al. 2007). In particular, due to its intuitiveness, the Douglas–Peucker algorithm (Douglas and Peucker 1973) has been used as one of the constituent techniques in a variety of procedures. It involves the use of a tolerance value

*Corresponding author. Email: drshishaozhong@gmail.com

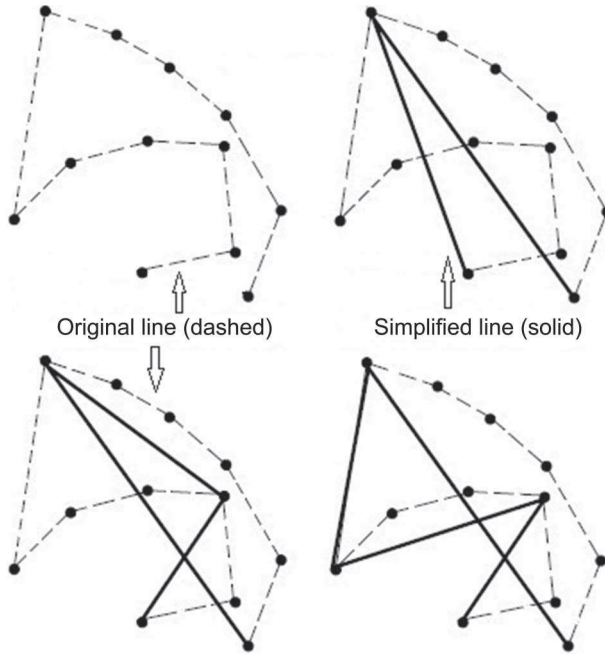


Figure 1. Synoptic illustration of the self-intersection effect caused by removal of anchoring points by the Douglas–Peucker algorithm with different tolerance values, when a line has a complex layout and turnings. The dashed line represents the original line, whereas the solid line represents a simplified line using less anchoring points (modified after Wu and Marquez [2003]).

as the constraint for decision-making for the removal of anchoring points of a line. However, existing automated procedures have serious shortcomings (Van Der Poorten and Jones 2010). Van Der Poorten and Jones (2010) stated that a failure common to many of the line simplification methods so far proposed is that they treat the real-world features as abstract geometric entities rather than lines with complex layout patterns (Figure 2). Map generalisation with line simplification remains as an interesting and open question. Further research is needed. A re-thinking of concepts, approaches, methods and procedure is needed to prevent problems of self-intersection (Figure 1) and to deal with boundary conflicts potentially arising from intersection associated with generalisation of real-world map data. There is a need to deal with post-generalisation gaps and overlaps to seamlessly match shared edges of adjacent polygons (Figure 3). This is needed in a situation when a great reduction of points needs to be made to area features and when seamless matching of shared boundaries between adjacent area features needs to be achieved.

The paper presents recent research and a procedural means to achieve the goal of automatic topologically consistent map generalisation. In addition, it describes the methods used for dealing with gaps and overlaps, should the edges of touching adjacent area features intersect rather than matching seamlessly.

The paper is organised as follows. Section 2 presents the new approach used. Section 3 describes methods which have been selected to achieve the goal. Section 4 provides the new overall procedure. Section 5 provides information on implementation and data sets used for experimentation, validation, testing and general performance of the program developed. Section 6 provides a comparison between the new procedure and the technique of the Douglas–Peucker algorithm. Section 7 concludes the paper.

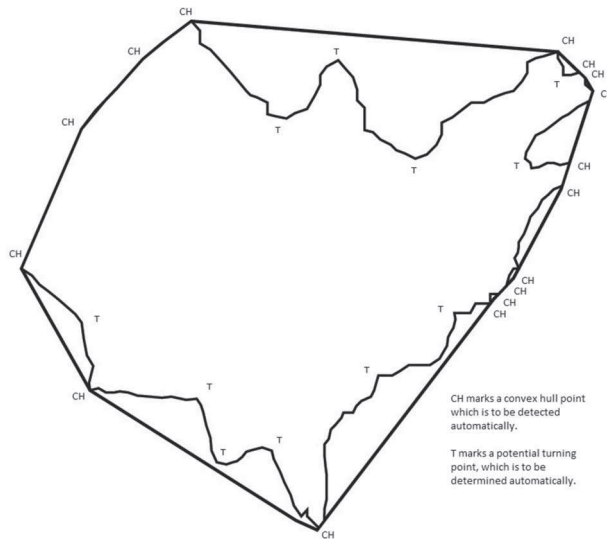


Figure 2. Synoptic illustration of a real-world complex polygon, its convex hull and turning points to be detected. Pivotal to partitioning of a complex line into monotonic chains is to find correct splitting points. Attention needs to be paid to the fact that combined use of turning points and convex hull points can allow partitioning of a complex line into monotonic chains. Line simplification with the Douglas–Peucker algorithm is confined within monotonic chains.

2. The approach

The approach used here employs turning points and convex hull points as a set of characteristic points to define the shape and characteristics of real-world geographical features with complex delineating line rendering layouts, many turnings and inherent sub-features. The set of characteristic points are used as splitting points to partition lines into monotonic chains. The characteristic points are automatically detected, recorded and always kept in the procedure. Line simplification with the Douglas–Peucker algorithm is confined within monotonic chains to guarantee no self-intersection. Once each monotonic chain is simplified by the removal of points using the Douglas–Peucker algorithm, the feature is re-composed with all characteristic points and simplified line chains. Once all features are simplified, a generalised map is re-composed with the simplified features. As there may be boundary conflicts arising from cross-intersection between touching adjacent area features, a post-generalisation processor is designed and programmed for detecting and solving each boundary conflict (gap or overlap) in turn. This research explored the use of the most suitable and intuitive means to deal with gaps and overlaps, which potentially result from cross-intersections, without adding or deleting any anchoring point. Boundary conflicts are due to the fact that delineating lines at shared edges no longer have identical anchoring points. A post-generalisation processor was produced to re-process these line parts using existing post-generalisation anchoring points in order to provide the delineating line parts with the identical lists of anchoring points. In this way, line parts at shared edges are re-formed to go through identical paths underpinned by identical lists of anchoring points. This edge-matching technique incurs neither spatial nor topological displacement of any anchoring point, and does not add or delete any anchoring point in the treatment of boundary conflicts.

The practical implementation of this approach is to decompose the map and obtain and save data on individual geographical features in the computer memory, generalise each

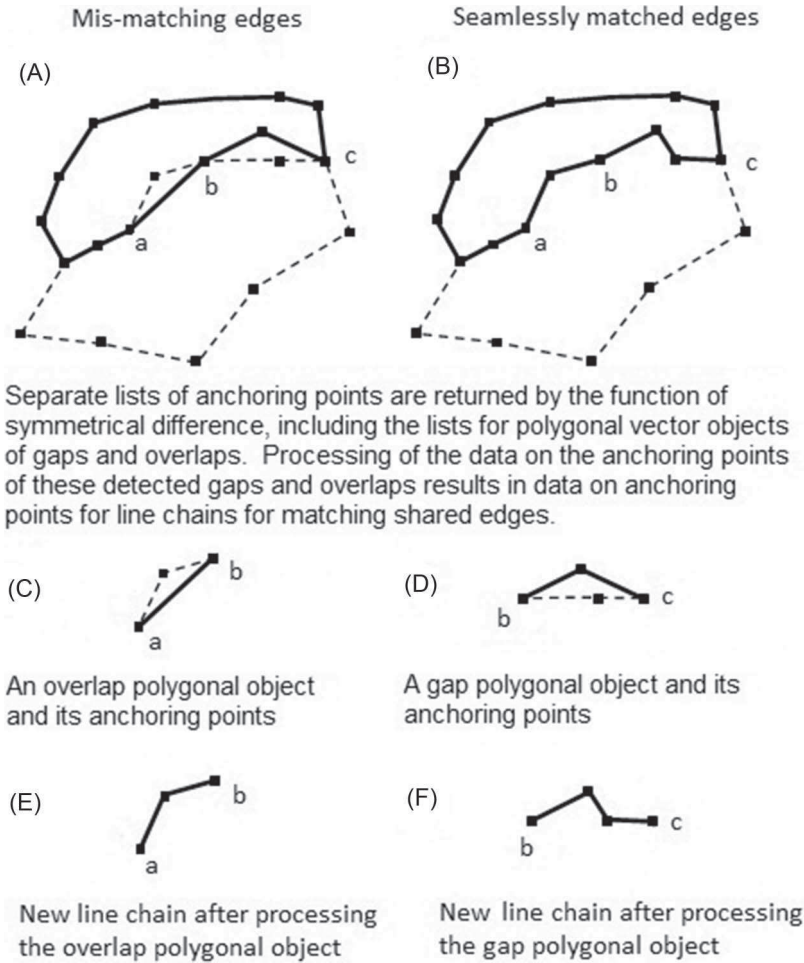


Figure 3. Synoptic illustration of two intersecting polygons, gaps and overlaps and associated anchoring points. Note that two adjacent polygons are not seamlessly matched between point a and c (Inset A). This is due to the fact that the adjacent polygons do not share identical lists of anchoring points between points a and c. An overlap occurs between points a and b (Inset A). A gap appears between points b and c (Inset A). In this case, separate lists of anchoring points are returned by the function of symmetrical difference of the JTS topology suite (Vividsolutions 2012), including the lists for polygonal vector objects of gaps (Inset D) and overlaps (Inset C). Processing of these anchoring points can produce new line chains (Insets E and F) using these anchoring points. The new line chains are used to replace the data where the overlap and gap appear. Use of identical list of anchoring points for shared edges of the two polygons effectively results in seamless edge matching (Inset B). It is important to note that there is neither spatial nor topological displacement of anchoring points. There is neither adding nor removing of any anchoring point with this method of edge matching.

feature and then recompose the map with the results. The treatment of each feature is made using the following methodology.

- Detect turning points and convex hull points
- Use these points as splitting points to partition the delineating line for each geographical feature into monotonic chains

- Line simplification with the Douglas–Peucker algorithm is applied to each monotonic chain to reduce the number of points used for the delineation of the line
- Recompose the geographic feature to form a generalised representation of the feature

Turning points and convex hull points are critical points for defining the characteristics of geographical features. Finding convex hull points effectively means locating the exterior-most points. Turning points and convex hull points are retained throughout the whole process to ensure that these points are always kept for the maintenance of the shape and characteristics of the features. As these points are used as the splitting points, they become the new starting and ending points of each monotonic line chain. The points are never removed in the process of point reduction when the technique of the Douglas–Peucker algorithm is applied. These splitting points are used as new starting and ending points for each monotonic chain. The application of the technique of the Douglas–Peucker algorithm only removes points between these splitting points (i.e. starting and ending points of each monotonic line chain). As each line chain between each pair of splitting points is monotonic in nature, it is guaranteed that there is no potential of any occurrence of self-intersection.

3. Description of methods

The combined use of selected methods includes the following.

3.1. Detection of turning points

Turning points were detected and recorded for each vector geographic information systems (GIS) object. The input of the turning point detector includes an ordered list of coordinates of anchoring points and a tolerance value to be applied. The output is a list of indices of detected turning points.

Whenever there is a significant change in the rendering direction of a line chain with respect to the given tolerance, a change in polarity is detected and the index of a turning point is recorded in a list. The detector processes anchoring points of the line in their original order. When the rendering of the line chain in one directional trend is within the given tolerance, the line chain is considered monotonic. Whenever there is an abrupt directional change in the layout of the line chain with respect to the given tolerance, a turning point is detected and recorded. At such a point, there is a polarity sign change indicating a turning point linking separate monotonic chains of the line (Figure 4).

3.2. Detection of convex hull points

Detection of Convex hull points is made using a Graham Scan (Graham 1972).

3.3. Mandatory safe-keeping of splitting points

The turning points and convex hull points constitute a set of splitting points, which define the characteristics of the geographical feature. A record list is used for mandatory safe-keeping of these points to ensure that the characteristics of the geographical feature are maintained.

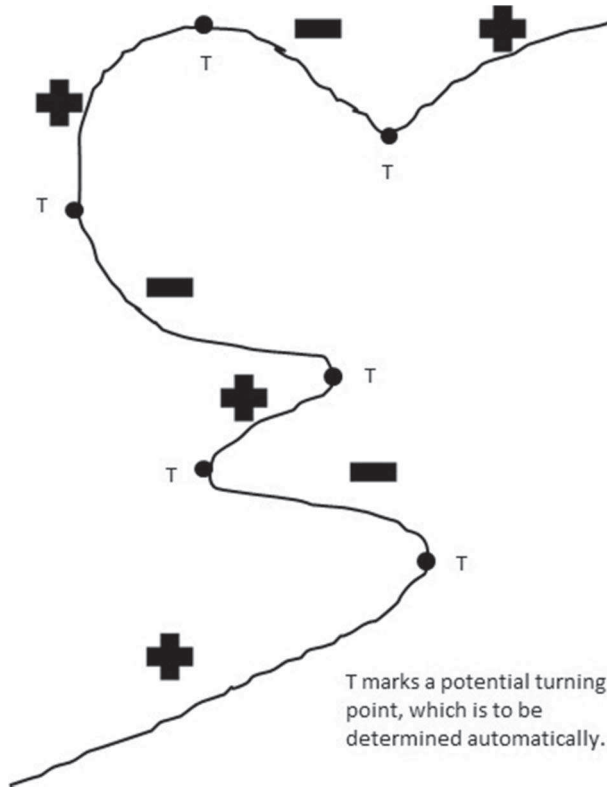


Figure 4. Synoptic illustration of a section of a real-world complex line with monotonic chains, layout polarity of chain and turning points.

3.4. *Simplification of each line chain with the Douglas–Peucker algorithm*

Although there are a range of alternative line simplification algorithms available, such as Christensen (1999), Li and Openshaw (1992) and Visvalingam and Whyatt (1993), the Douglas–Peucker algorithm has been chosen and implemented as a nested function to carry out line simplification of each line chain, as it is a widely used and efficient point reduction technique. In addition, the Douglas–Peucker algorithm was chosen as it offers intuitive simplicity as a point reduction technique based on the input of a tolerance value. Therefore, the point reduction is confined to each individual line chain in this research by applying the same tolerance value, which is used for detecting turning points.

3.5. *Re-composition of generalised features*

A generalised feature is re-composed using all simplified line chains for the feature.

3.6. *Seamlessly matching shared edges between touching adjacent area features*

This is designed to deal with situations when inevitable gaps and overlaps occur as the result of cross-intersections of the edges of touching adjacent area features, if a great reduction of points is made. The perceptible intersections of line parts between

touching adjacent area features yield gaps and overlaps. The class of the Symmetrical Difference of Java Topology Suite (Vividsolutions 2012) (*a.k.a* JTS) is used as the surrogate detector of gaps and overlaps. The method returns data of anchoring points of gaps and overlaps in a manageable and structured way. Gaps and overlaps are caused by the fact that area features do not have identical lists of anchoring points at places of shared edges. In this research, the function of symmetrical difference is used as the surrogate detector of gaps and overlaps. Based on this, Java classes were designed and implemented for processing detected gaps and overlaps with the JTS's Symmetrical Difference class. This effectively results in producing identical lists of anchoring points between shared edges, which replace the anchoring points of parts of the line chain with the anchoring points of the gaps and overlaps. This effectively results in seamless matching of shared edges at places where gaps and overlaps occur after reduction of points in the simplification process, which is a form of edge matching using existing anchoring points after the simplification process, without adding or deleting any anchoring point. This is to say that identical lists of anchoring points that are produced using this post-generalisation processing are used as replacement anchoring points of line parts at places where gaps and overlaps occur. This allows seamless matching of edges of lines to be achieved.

4. Procedure

The procedure has two parts. One is for topologically consistent map generalisation. The other is for seamless matching of shared edges of touching adjacent area features.

4.1. *Topologically consistent map generalisation*

By applying the methods described above, the procedure was created using the following steps.

- Decompose map data set and save in an array of data of anchoring points of lines.
- Treat area features as closed lines
- Generalise each line by detecting and recording turning points and convex hull points.
- Use turning points and convex hull points as splitting points.
- Split the line into monotonic chains.
- Simplify each monotonic line chain with the Douglas–Peucker algorithm.
- Re-compose the feature with turning points, convex hull points and simplified line chains.
- Re-compose the map with all generalised features and output the map as a Shapefile (ESRI 1998).

4.2. *Seamless matching of touching adjacent area features*

This has been implemented as a post-generalisation processor to ensure that no output of gaps and overlaps will occur in the output map (Figure 5). When a great point reduction is made, there will inevitably be gaps and overlaps between touching adjacent area features. This post-generalisation processor is designed to remove gaps and overlaps without the addition or deletion of any anchoring point. The purpose of the processor is to make sure

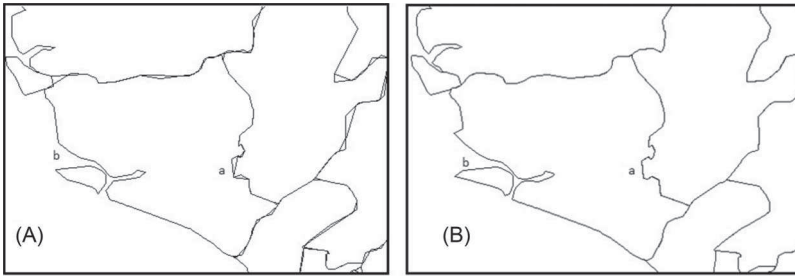


Figure 5. An example section of vector-based map containing intersections, resultant gaps and overlaps when the technique of the Douglas–Peucker algorithm is applied to each monotonic line chain (A) and the result of automatic seamless edge matching produced by the new post-generalisation processor (B). Note that only touching adjacent polygons are seamlessly edge-matched (a). Not touching adjacent polygons are left unprocessed (b).

that touching adjacent area features to use identical lists of anchoring points for delineating line parts at shared edges. This solution offers seamlessly matched edges for touching adjacent area features.

5. Implementation, data sets tested and performance

5.1. Data sets tested and formats of input and output

A Java program was designed and programmed by implementing the methods and procedure described in this paper and a set of Java classes was written for the implementation of the methods. The actions of classes were coherently chained together to realise the whole program. The program works as both a stand-alone software and an on-demand server-side processor which can be called to work as a backbone geodata processor by Internet Mapping services. The program reads in map data in the form of the ESRI's ShapeFile format and outputs the final map data in either the ESRI's ShapeFile format or CSV format.

The program was implemented on a Dell laptop with an Intel 2.8 GHz dual core processor and 8 Gbytes of main memory running 64-bit Windows Operating System. The program has been iteratively experimented, validated and tested.

Automatic processing for map generalisation of digital map data sets of the United Kingdom (the total number of points in the original is 903123 and the total number of points in the file of the generalised map is 894001) and USA boundaries (the total number of anchoring points in the original file is 810539 and the total number of points in the file of the generalised map is 806295) was made with the approach and method presented in this paper. The duration of processing of these files (World Countries, UK boundaries 22 Mbytes and USA boundaries 13 MBytes) takes approximately 2 minutes. The results of neither gaps nor overlaps can reach approximately 30% of data compression. This was achieved by finding and using appropriate tolerance values through experimenting with the input of a range of different values (Figure 6). The maximum point reduction is dependent on the data set and has been observed to reach approximately 40%, but gaps and overlaps occur. The seamless edge matching removes the gaps and overlaps for touching adjacent polygons, by making the shared edges use identical lists of anchoring points.

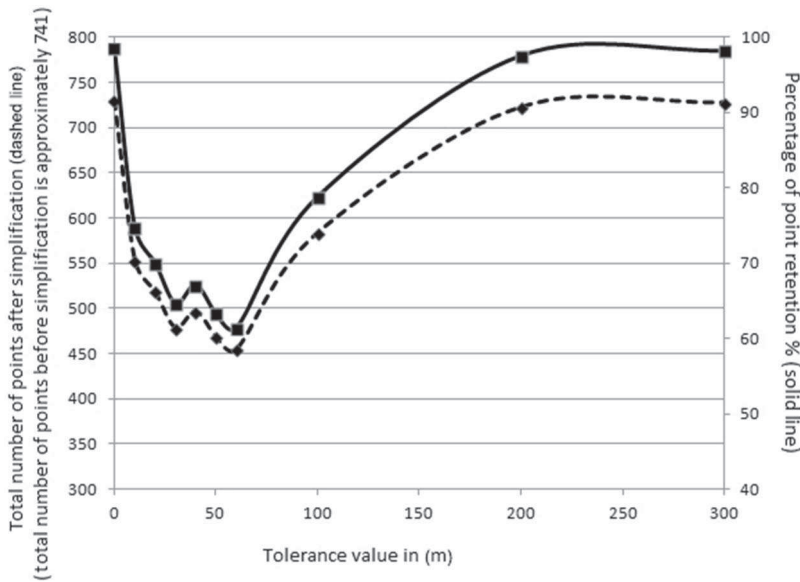


Figure 6. Effect of application of the programmatic solution to a selected testing data set of Dublin Enumeration Districts with different tolerance values. Testing on data sets of the World Country Map, US State boundaries and UK regional boundaries shows the same effect and very similar curves. Note: The horizontal axis is the tolerance value applied. The left vertical axis indicates the total number of points after map generalisation. The total number of points before map generalisation is 730. The left axis indicates the percentage of point retention. Starting from 0, with an increase of tolerance value, the percentage of point reduction increases. When the tolerance value increases further from a threshold value, the percentage of point retention starts a rising trend, as more turning points are detected.

6. Comparison between the new procedure and the technique of the Douglas–Peucker algorithm

The application of the Douglas–Peucker algorithm may cause self-intersection (Figure 1) and intersection (Inset A of Figure 3 and Inset A of Figure 5), when it is applied on its own (e.g. [1], [13]). Intersections yield gaps and overlaps (Inset A of Figure 3 and Inset A of Figure 5). The new procedure guarantees outputting neither self-intersection nor intersection (Inset B of Figure 5). The procedure outputs neatly presented generalised maps (Inset B of Figure 5) and has two parts. Once a map is decomposed, individual features are stored centrally in the memory. The first procedure is to generalise each feature in turn and the generalisation may yield intersections even when no occurrence of self-intersection is guaranteed (Inset A of Figure 5). The second procedure is to detect and solve each gap or overlap in sequence and to guarantee seamlessly matched edges for the final output (Inset B of Figure 5).

7. Conclusions

In this paper, we have described a new approach using selected intuitive methods to realise a novel procedure for automatically generalising maps which depict real-world geographical phenomena. This procedure offers the direct processing of vector-based geographical data sets. This procedure maintains the characteristics of geographical

features guaranteeing outputs that neither self-intersect nor cross-intersect. The results of the iterative process of exploratory programming, experimentation, validation and testing of the new program demonstrate that it is capable of generalising map features composed of complex layout patterns, many turnings and inherent sub-features. Thus, the program is appropriate for the automatic generalisation of maps that depict real-world phenomena.

Acknowledgements

The authors are grateful to Fernando Branco for valuable discussions over the persistent problems encountered and Yu Chen and Tim O'Farrell for technical assistance at the preliminary stage of program development. The authors thank Dr [Gavin McArdle](#) and Dr Eoghan Furey of NUIM, Professor David Smith of Oxford University and Professor [Jim Hansom](#) of Glasgow University for proofreading. Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

- Christensen, A. H. J. 1999. "Cartographic Line Generalization with Waterlines and Medial-Axes." *Cartography and Geographic Information Science* 26 (1): 19–32.
- Corcoran, P., P. Mooney, and A. Winstanley. 2011. "Planar and Nonplanar Topologically Consistent Vector Map Simplification." *International Journal of Geographical Information Science* 25 (10): 1659–1680.
- Da Silva, A. C. G., and S. T. Wu. 2005. "Preserving Coincidence and Incidence Topologies in Saalfeld's Polyline Simplification Algorithm." Accessed July 5, 2012. www.geoinfo.info/geoinfo2005/papers/p64.pdf
- Douglas, D. H., and T. K. Peucker. 1973. "Algorithms for the Reduction of the Number of Points Required to Represent a Line or Its Caricature." *The Canadian Cartographer* 10 (2): 112–122.
- ESRI. 1998. ESRI Shapefile Technical Description, An ESRI White Paper – July 1998. Accessed August 10, 2012. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
- Graham, R. L. 1972. "An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set." *Information Processing Letters* 1: 132–133.
- Li, Z., and S. Openshaw. 1992. "Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization." *International Journal of Geographical Information Systems* 6 (5): 373–389.
- Mantler, A., and J. Snoeyink. 2000. "Safesets Line Simplification." Accessed June 12, 2012. <http://www.cs.unc.edu/~mantlers/research/safesimp/proof2.ps>
- Saalfeld, A. 1999. "Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm." *Cartography and Geographic Information Science* 26 (1): 7–18.
- Van Der Poorten, P. M., and C. B. Jones. 2010. "Characterisation and Generalisation of Cartographic Lines Using Delaunay Triangulation." *International Journal of Geographical Information Science* 16 (8): 773–794.
- Visvalingam, M., and J. D. Whyatt. 1993. "Line Generalisation by Repeated Elimination of Points." *Cartographic Journal* 30 (1): 46–51.
- Vividsolutions. 2012. "JTS Topology Suite." Accessed August 10, 2012. <http://www.vividsolutions.com/jts/jtshome.htm>
- Wu, S. T., and M. G. Marquez. 2003. "A Non-Self-Intersection Douglas-Peucker Algorithm." *Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'03)*, Sao Carlos, October 12–15.
- Zhang, C., M. Pan, H. Wu, and H. Xu. 2007. "Study on Simplification of Contour Lines Preserving Topological Coherence." *Acta Scientiarum Naturalium Universitatis Pekinensis* 43 (2): 216–222.