GSR_2
School of Mathematical and Geospatial Sciences, RMIT University
Monday 10 to Wednesday 12 December 2012

Polygon Generalization with Circle Arcs

Antoni Moore[1], Chris Mason[2], Peter Whigham[3] and Michelle Thompson-Fawcett[4]

[1]School of Surveying / [2]Department of Anatomy and Structural Biology / [3]Department of Information Science / [4]Department of Geography, University of Otago, New Zealand

tony.moore@otago.ac.nz

Abstract
Despite the current high capacity and speed of computers, the efficient storage of spatial data is still a cutting edge issue, most notably in the context of mobile devices. Processing power, limited storage and small display on mobile devices all mean that algorithms which efficiently summarize spatial data, reducing its size, have relevance. Generalization also has an important role to play in mobile display, not merely being employed for scale change, but overall legibility.

This paper investigates the accuracy of using circles to store polygon boundaries. Can a series of xy points be usefully generalized by information contained in a smaller array of variably-sized circles used in a non space filling sense to approximate to the edge of the polygon? Accordingly, a Voronoi-based medial axis approach was used to generalize a vector dataset representing the island of Rarotonga.

Two measures were combined to ascertain the effectiveness of this, size of generalized dataset and visual error. Circle approximation was not found to outperform the state-of-the-art Douglas-Peucker generalization algorithm in terms of dataset size and visual accuracy, though suitability for modelling rounded coastlines and other like geographic features was highlighted and future research directions suggested.

Keywords: Visual Error, Spatial Data Storage, Cartographic Generalization

Biography
Antoni Moore is a senior lecturer in Geographic Information Science at the National School of Surveying, University of Otago, New Zealand. His research interests include geovisualisation, which encompasses the visualisation of uncertainty, cognitive mapping and application of virtual / augmented reality. Other research interests cover cartographic generalisation, spatial data structures and use of GIS-related technology in a decision support context. He was previously a lecturer in Otago's Department of Information Science from 2001-2007 and before that a coastal / marine GIS Analyst at Plymouth Marine Laboratory in the UK.

Introduction
Despite the current high capacity and speed of computers, the efficient storage of spatial data is still a cutting edge issue, most notably in the context of mobile devices. Processing power, limited storage and small display on mobile devices all mean that algorithms which efficiently summarize spatial data, reducing its size, have relevance. Generalization also has an important role to play in mobile display, not merely being employed for scale change, but overall legibility (Anand et al, 2008; Jones and Ware, 2005).

This paper reports on a novel and potentially useful spatial data generalization method and its initial testing for efficiency of storage and accuracy. The theory underlying this method is that the conventional storage of polygonal data in terms of a series of xy points can be efficiently replaced by an array of variably-sized circles that approximate closely to a polygon boundary (accuracy tests will measure how closely) and run in a non-space filling mode. The circle array holding the generalized line information implicitly has three values per entry: *x*, *y* and *r*, where *x*, *y* = the centre coordinates, and *r* = radius of the circle (as opposed to just *x*, *y* for conventional point-delineated polygon structures).

Methods

The circular arcs are derived through the following process.

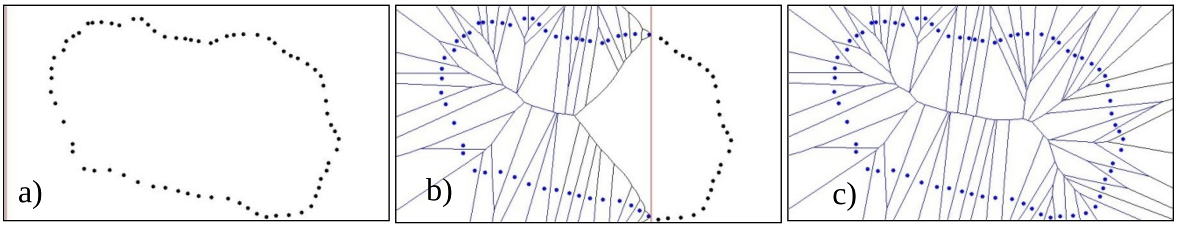
Deriving a population of circles through the sweep line Voronoi method and medial axes

The sweep line algorithm (Fortune, 1987) was used to generate a Voronoi diagram (Figure 1) which in turn described the medial axis of the polygon (that set of lines in the polygon interior describing a "skeleton" that is central to the polygon). One of the main advantages of using this algorithm was the automatic generation of a population of circles whose circles are coincident with the vertices of the Voronoi diagram.

In similar research, a union-of-circles based approach to shape representation was implemented by Ranjan and Fournier (1996) as the basis for assessing the similarity of two shapes as well as interpolation between them. Hubbard (1996) used 3D shape filling spheres as the basis for quickly calculating collision detection of shapes in a dynamic environment. Within the cartographic generalization research community Gold and Thibault (2001) and Haunert and Sester (2004) have applied medial axes or skeletons to the generalization task.

In the current implementation, each circle's radius was calculated during the execution of the algorithm and required minimal post processing upon completion. A full population of circles approximating to the polygon boundary was generated from the medial axis. The points of the original polygon are stored with the circle that approximates to them.

Figure 1. Stages in the application of the sweep line algorithm on point data. a) points delineating polygon data; b) sweep line triggers parabolas for each point; where parabolas meet a Voronoi boundary is formed (the full parabola for any point is not seen, being clipped at any Voronoi boundary); c) the final Voronoi diagram (generated by software in Odgaard and Nielsen, 2000).



Filtering

Four stages of filtering were then needed to remove unwanted circles. The first stage takes an error threshold, epsilon, and removes circles whose centres lie within an epsilon band ε (Perkal, 1956) around the boundary, as illustrated in Figure 2. This is the Minimum Circle Threshold (MCT), as reported in results.

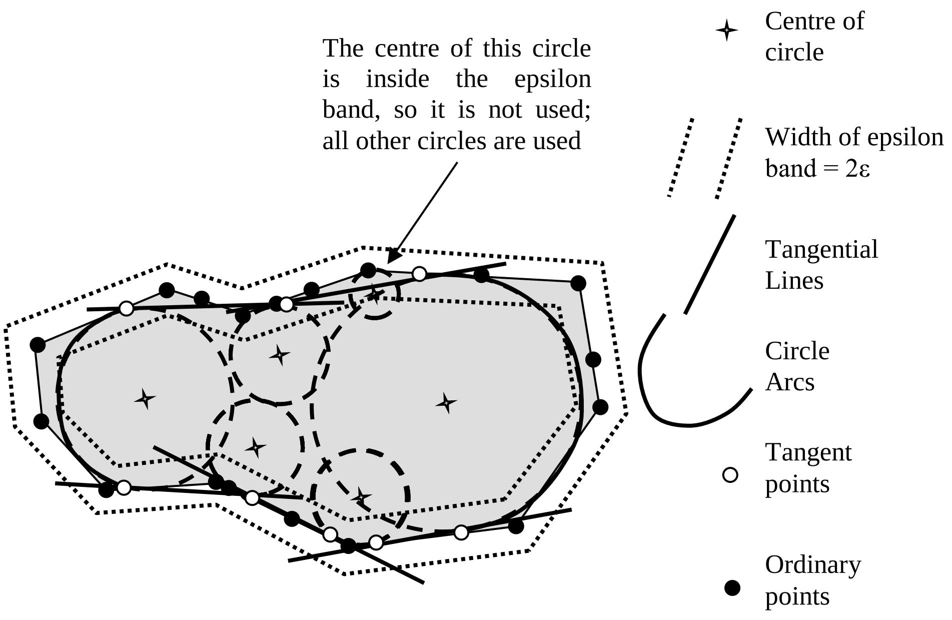
The second stage takes an overlap threshold *t* and removes circles whose overlap with smaller circles exceeds *t*. Circles are treated from smallest to largest so that detail contributed by the smaller circles is more likely to be retained. The third stage takes an error threshold and removes circles whose arcs extend more than that threshold outside the boundary. The fourth stage takes a minimum run length (threshold number of consecutive points on the boundary "captured" by a circle for that circle to be included) and maximum allowed gap, and removes circles that replace less than run length number of points on the boundary, as long as no gap of more than the maximum allowed gap is formed. This is calculated from assessment of the points already stored with the circle. The resultant list of circles is stored in the same order as the original polygon points.

Reconstruction of the polygon from stored circles

The conventionally stored circles (*x*, *y*, *r*) are internal circles, which approximate to the polygon boundary from the inside. As natural geographical objects are concave as well as convex, external circles, stored as (*x*, *y*, -*r*) or with negative radii are used. As Figure 2 shows (though external circles are not displayed here), the reconstructed polygon comprises a combination of tangents and circle arcs. The circles are stored in clockwise order in a linked list and are consecutively processed two at a time for construction of tangents. Where two consecutive tangents intersect, the coordinate of intersection forms a new point. All remaining gaps to be reconstructed are filled in by circle arcs. Finally, there is also a check for self intersection, with resultant loops removed at this stage.

* Review Paper – accepted after double-blind review.
ISBN: 978-0-9872527-1-5
Melbourne: Geospatial Sciences, RMIT 2012

Figure 2. Reconstructing the polygon using tangents and circle arcs



Testing

All realisations from this implementation were compared with output from the Douglas-Peucker algorithm (1973). This commonly used generalization method is global and recursive, initially offering the simplest straight line linking the start and end points of the line to be generalised. It then identifies the point that has the greatest perpendicular distance from the initial line, then subsequently uses that point to define two straight lines: start – point and point – end. The perpendicular distance measurement is then repeated recursively on these two smaller lines, and so on. Points from the original line are added to the evolving generalised line by the greatest perpendicular distance criterion until some predefined threshold (minimum allowed perpendicular distance) is undercut. This threshold thus effectively defines the scale of generalization.

The Douglas-Peucker algorithm has been proven to be effective at retaining the most important points in defining the recognisable shape of a line. These tend to be the most extreme points, and these alone can result in unrealistic spiky artefacts in the generalized line, particularly at higher thresholds. The method proposed in this paper adopts a smoother, curve-based solution to address this.

There are many generalization algorithms that have been put forward over the years, and it is not the place of this paper to compare them. A selection is cited here for affinity with the technique proposed. For example, Saux (2003) uses B-splines and wavelets, which have curved geometry; Wang and Muller (1998) use curves too. Christensen (2000) uses medial axes as part of his generalization solution.

Error measurements of the realisations took two forms which were subsequently combined for the results. Firstly, the amount of data stored for a realisation was recorded, in terms of a count of numbers needed for storage (in the case of each circle, three: *x*,*y*,*r*; two numbers in the case of points chosen by Douglas-Peucker). Secondly, measurement of the visual area as used by Alani et al (2001) was implemented. This is a graphic measure, assessing parity of shape. It is given by the formula

$$(A_{pp} + A_{np}) / \text{actualArea} \quad (1)$$

where A_{pp} is the positive approximate false error (area falling within the approximation but outside of the original) and A_{np} is the negative equivalent (area falling within the original but outside of the approximation). In this study, the number of points (indicating performance in data storage) and visual error (ranging from 0 to 1) will be multiplied to give an overall figure of effectiveness for each realization.

* Review Paper – accepted after double-blind review.
ISBN: 978-0-9872527-1-5
Melbourne: Geospatial Sciences, RMIT 2012

Results

The circle-packing algorithm has been tested on a 1:62800 polygon of the South Pacific island of Rarotonga, chosen for its smoothness and a favourable initial test. Many combinations of parameters were tried in order to explore the properties of the algorithm, producing many realisations (sample output in Figure 3).

Figure 3. a) the original Rarotonga polygon; (b) an example Douglas-Peucker generalization output (100m); (c) a representation derived through altering MCT = 25, overlap = 90%, minimum 3 point runs with no gaps; (d) very generalized representation, the result of MCT = 100, overlap = 80%, runs of 4 points and above with at most gaps of one point permitted.

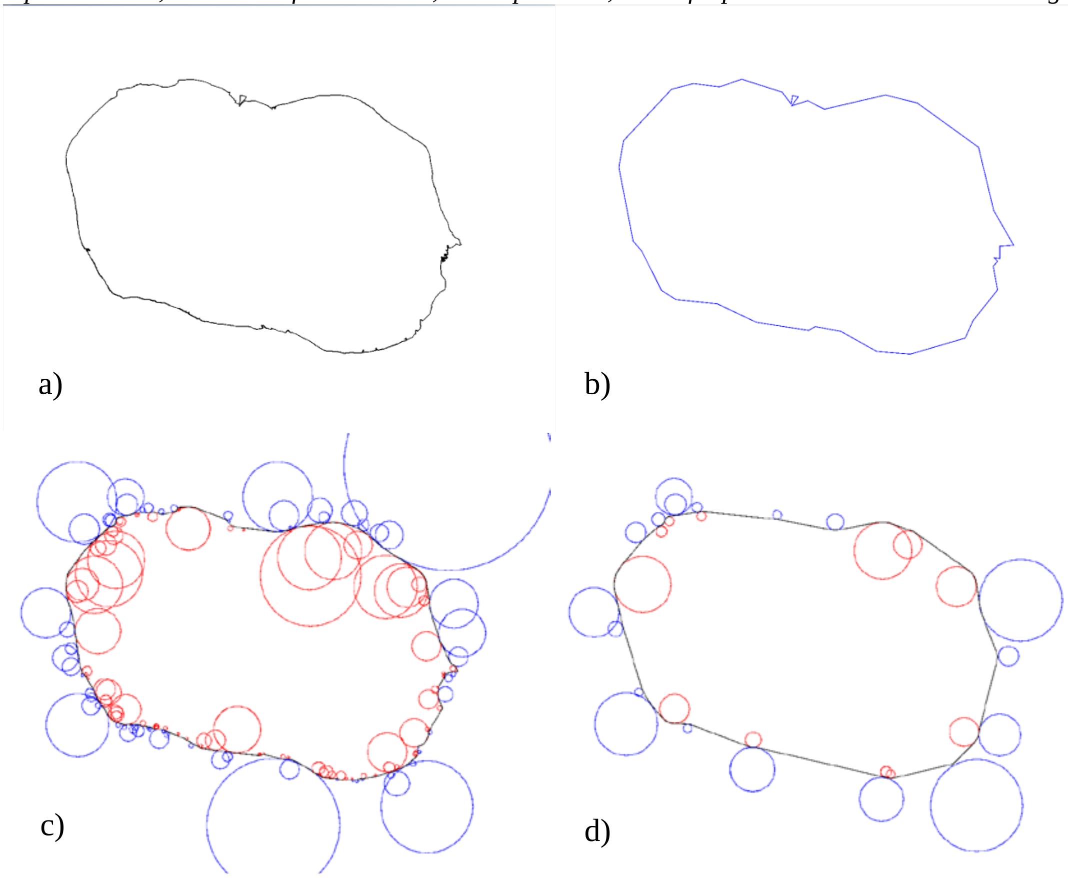
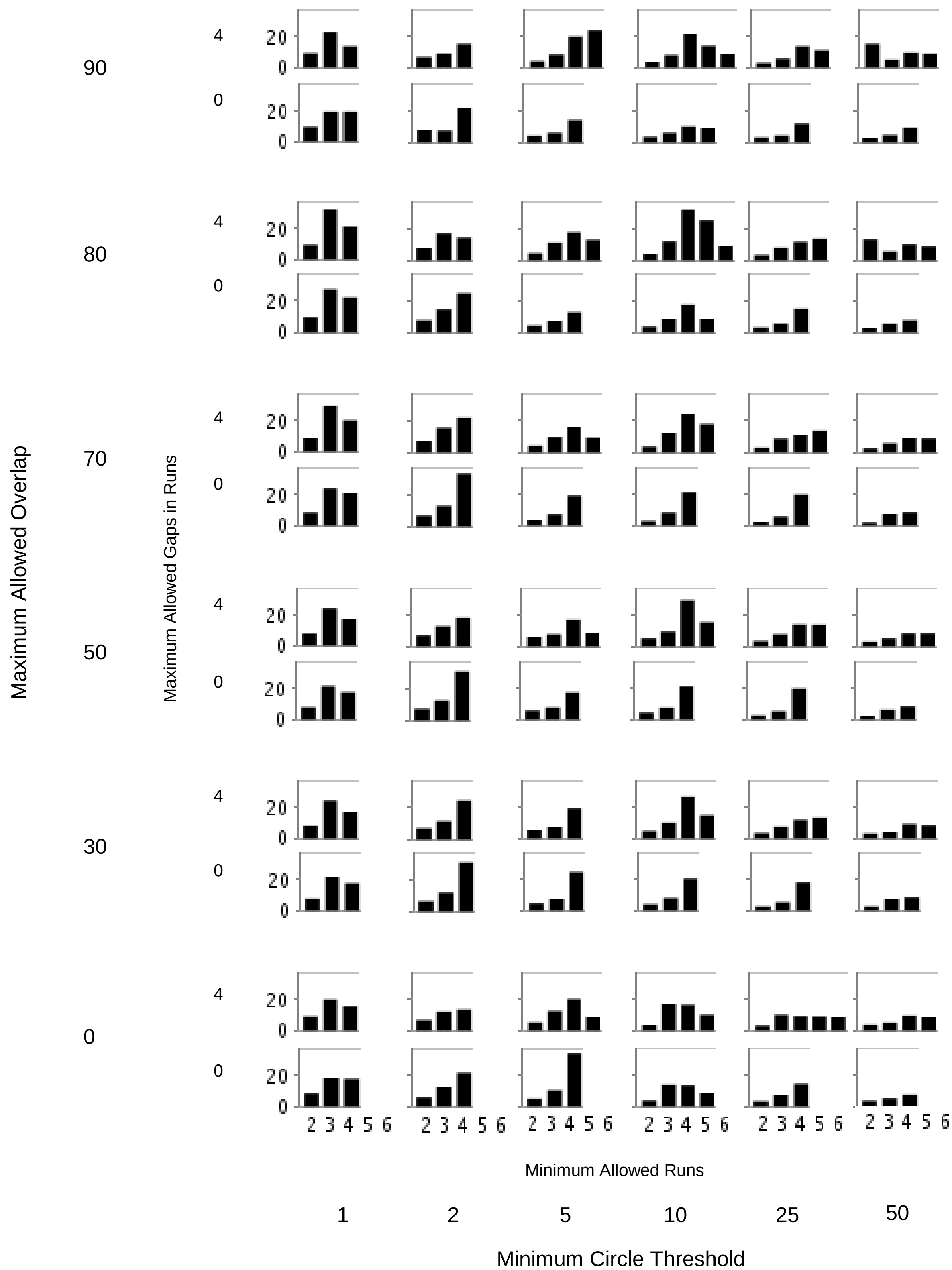


Fig. 4 graphs the error measurement scores of a representative selection of realizations. Values graphed for the parameters were chosen as follows: The Minimum Circle Threshold scale was relative to the mean point separation of the original dataset; an even spread of values was graphed for Maximum Allowed % Overlap; Minimum Allowed Runs was displayed for the range of values (2 up to 6) yielding results; and Maximum Allowed Gaps values chosen (0, 4) were sufficiently separated to show some difference in error magnitude.

Looking for patterns within and between realizations, there seems to be no benefit in lifting the minimum points in a run. However, there is an abrupt improvement in performance with increasing minimum circle threshold (markedly improved results once MCT= 50, perhaps a reflection of mean point separation of the original Rarotonga dataset). Fig. 5 graphs the scores for the Douglas-Peucker output. It can be seen from the measures that Douglas–Peucker outperforms circle-based generalization (a lower score means low error and low number of points – from this measure, DP is over twice as effective as circle approximation at its best).

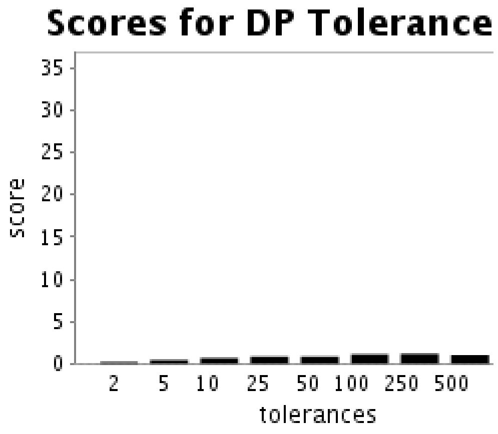
* Review Paper – accepted after double-blind review.
ISBN: 978-0-9872527-1-5
Melbourne: Geospatial Sciences, RMIT 2012

*Figure 4. The (Number of points * Visual Area) values graphed by Minimum Allowed Threshold, Maximum Allowed Overlap, Minimum Allowed Runs and Maximum Allowed Gaps in Runs.*



* Review Paper – accepted after double-blind review.
ISBN: 978-0-9872527-1-5

Figure 5. The error values graphed for different Douglas Peucker tolerances.



Discussion and Conclusion

From a polygon of the island of Rarotonga, the circle algorithm has derived a set of circles, whose arcs approximately follow the coastline, but is smaller in size than the original data set. However, even if the algorithm could reconstruct polygons perfectly from circles, there is still the issue of the loss of explicit boundary coordinates – storage is in terms of circle centres, which are generally placed somewhat remote from the boundary.

Beyond these initial results, more challenging and intricate polygon forms (i.e. containing convexities and concavities in particular of different scales) will be used to test the boundary approximating circle algorithm, to further explore its properties. Although not outperforming Douglas-Peucker, the kind of curved coastlines being generated by the circle algorithm may be a step towards the 'aesthetic' generalization looked for by Dutton (1999) (and user testing could be applied to state this for definite). Other possible future strategies include running the circles in space-filling mode (enabling swift containment calculations), exploring circle ordering (largest first as opposed to smallest first) and optimization techniques. Finally, to ascertain efficiency of the proposed algorithm (critical in the context of mobile devices) an assessment of processing times will be made.

Acknowledgements

This research was supported by a University of Otago research grant. We would like to thank Darrin Drumm for providing the Rarotonga coastline data. We would also like to thank the reviewers for their feedback.

References

Alani, H, Jones, C B and Tudhope, D. (2001). Voronoi-based region approximation for geographical information system retrieval with gazetteers. *International Journal of Geographical Information Science*, 15, 4, 287-306.

Anand, S, Ware, J M and Taylor, G. (2008) Generalisation of Large-Scale Digital Geographic Datasets for MobileGIS Applications. In: J Drummond, R Billen, E João and D Forrest (eds.) *Dynamic and Mobile GIS*. CRC Press, Boca Raton.

Christensen, A H J. (2000) Line generalization by waterlining and medial-axis transformation. Successes and issues in an implementation of Perkal's proposal. *Cartographic Journal*, **37**, 1, 19-28.

Douglas, D and Peucker, T. (1973). Algorithm for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10, 112-122.

Dutton, G (1999) Scale, Sinuosity and Point Selection in Digital Line Generalization. *Cartography and Geographic Information Science*, 26, 1, 33-54.

Fortune, S. (1987). A Sweepline Algorithm for Voronoi Diagrams. *Algorithmica*, 2, 153-174.

Gold, C and Thibault, D (2001) Map Generalization by Skeleton Retraction. *ICA Workshop on Generalisation and Multiple Representation, 12th to 14th August 1999, Ottawa, Canada*.

Haunert, J-H and Sester, M (2004) Using the Straight Skeleton for Generalisation in a Multiple Representation Environment. *ICA Workshop on Generalisation and Multiple Representation, 20th to 21st August 2004, Leicester, UK.*

Hubbard, P. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15, 3, 179-210.

Jones, C B and Ware, J M. (2005) Map Generalization in the Web Age. *International Journal of Geographical Information Science*, 19, 8/9, 859-870.

Odgaard, A and Nielsen, B K. (2000). *A visual implementation of Fortune's Voronoi algorithm.* http://www.diku.dk/hjemmesider/studerende/duff/Fortune. Accessed: 4th May 2012.

Perkal, J. (1956). On epsilon length. *Bulletin de l'Academie Polonaise des Sciences*, 4, 399-403.

Ranjan, V and Fournier, A (1996). Matching and Interpolation of Shapes using Unions of Circles. *Computer Graphics Forum*, 15, 3, 129-142.

Saux, E. (2003) B-spline Functions and Wavelets for Cartographic Line Generalization. *Cartography and Geographic Information Science*, 30, 1, 33-50.

Wang, Z and Muller, J-C (1998) Line Generalization Based on Analysis of Shape Characteristics. *Cartography and Geographic Information Systems*, 25, 1, 3-15.

* Review Paper – accepted after double-blind review.
ISBN: 978-0-9872527-1-5
Melbourne: Geospatial Sciences, RMIT 2012