The storage and reconstruction of polygon boundaries using circles

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

[1]Spatial Information Research Centre, Department of Information Science, /
[2]Department of Geography, University of Otago,
PO Box 56, Dunedin, New Zealand.
Phone: +64 3 4798138 Fax: +64 3 4798311;
Email: amoore@infoscience.otago.ac.nz

Presented at SIRC 2003 – The 15th Annual Colloquium of the Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
December 1st & 2nd 2003

ABSTRACT

This paper investigates the accuracy of using circles to store polygon boundaries. Can a series of xy points be accurately replaced by a smaller array of variably-sized circles used in a non space filling sense to approximate to the edge of the polygon? Two measures were used to ascertain this, areal error and visual error.
A vector dataset representing the island of Rarotonga was converted from boundary coordinates to circle storage using a Voronoi-based medial axis approach. Three parameters were altered to derive as many as 150 circle-based realisations – minimum allowed circle radius, degree of overlap allowed between circles and Douglas-Peucker algorithm threshold (the latter algorithm of line reduction was used to extract "important" points from the polygon and add them to the circle list as "zero-radius circles"). This presentation presents and analyses the areal error and visual error results for all realisations.

***Keywords and phrases*:** Areal error, visual error, cartographic generalisation

1.0 INTRODUCTION

Despite the current high capacity and speed of computers, the storage of spatial data is still a cutting edge issue, most notably in the context of the Internet (Wu and Amaratunga, 2003). Also a factor to a lesser extent are widely-accessible small screen appliances. Market studies indicate that the number of these devices will grow rapidly in the future, yet they have low memory and poor CPU performance (Pham *et al* 2000).

This paper is concerned with a method that promises to be an efficient and accurate spatial data storage means. The theory 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 to a polygon boundary and run in a non-space filling mode (one of the assumptions is that there is a threshold minimum circle; any areas smaller than this cannot have a circle occupying them). The method used was developed in the context of a circle tree (Moore, 2002), a hierarchical data structure with the anticipated properties of efficient storage, fast access and multiscale representation of spatial data. Issues to do with tree structure and multi-scale representation will not be covered in this part of the project; indeed the algorithm used here assumes no tree structure.

The circle array 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 the competing point-delineated polygon structure). The theory behind the circle tree draws from the fields of computer graphics (collision detection of 3D objects – Hubbard, 1996), databases (indexing through the R-tree – described in Rigaux *et al*. 2002 – and sphere tree structures – van Oosterom, 1993) and cartography (e.g. multiscale generalisation through use of the Douglas-Peucker algorithm – Jones and Abraham, 1987). A more in-depth overview of the background literature is given in Moore (2002).

The next section of the paper briefly describes the algorithm employed to derive a list of circles from a polygon boundary (using the polygon's medial axis, which in turn is extracted from the Voronoi

diagram of the polygon vertices) and associated issues such as filtering, internal / external circles, and reconstructing the polygon from circles. In deriving the set of circles, three parameters can be changed: minimum circle size, degree of overlap between circles and Douglas-Peucker algorithm threshold (key coordinates are stored as "zero-radius circles"). The Douglas-Peucker algorithm (Douglas and Peucker, 1973) is the line reduction method of choice for many commercial GIS, owing much to its ability to retain the essential characteristics of a geographic object when generalising (i.e. it retains the important points – Jones, 1997). For this reason it is used in the course of this research.

A vector dataset of the island of Rarotonga will be processed, producing many circle-based approximations to the original data (a product of extensively using various combinations of the three parameters). These approximations will be tested for accuracy using the areal error and visual error tests used by Alani et al (2003). The results will be shown in map and graph form, before discussion and concluding remarks.

2.0 METHODS

2.1 Formal Description

It makes sense to start the algorithm description with an account of the assumptions made. The input data will describe standard spatial polygons of the form (x_1, y_1), (x_2, y_2),…,(x_n, y_n), where (n-1) = number of points comprising the polygon and $(x_1, y_1) = (x_n, y_n)$ to close the polygon. The aim of the circle-packing algorithm is to replace such a set of points with a smaller set of spatially distributed circles of differing size. This series of circles may be represented in this way: (x_1, y_1, r_1), (x_2, y_2, r_2),…,(x_m, y_m, r_m) where x,y are the coordinates of the circle centre, r is the radius of the circle and m = number of circles used to approximate to the polygon. Unlike the point representation, (x_1, y_1, r_1) need not equal (x_m, y_m, r_m). It is the arcs of these circles, and the tangents linking two adjacent circles, that form the new basis of polygon boundary representation, to within an acceptable accuracy, as described in section 2.4.

2.2 The plane sweep Voronoi method and medial axes

Like Hubbard's (1996) sphere tree method, the circle algorithm uses medial axes (or the 'skeleton' of the polygon) to help determine the location and size of circles that can effectively replace a few original polygon points. Medial axes can be generated from the boundaries of Voronoi polygons, which are calculated from the original polygon vertices (all area inside a Voronoi polygon is closer to the vertex that "owns" it than any other point). In this case, the plane sweep algorithm (Fortune, 1987) is used to compute the Voronoi diagram (a collection of Voronoi polygons). Figure 1 exemplifies the plane sweep process.

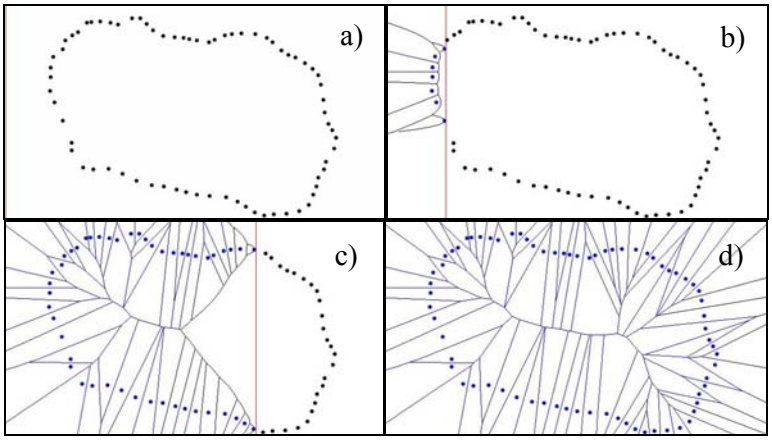

Figure 1. Stages in the application of the Fortune algorithm on point data. a) points delineating polygon data; b) sweep line triggers parabolas for each point; where parabolas meet a Voronoi boundary is formed; c) a later stage in the operation; d) the final Voronoi diagram (generated with software in Odgaard and Nielsen, 2003).

One of the main advantages of using the plane sweep algorithm was the automatic generation of a population of circles whose centres were formed at the Voronoi diagram vertices. Each circle's radius is calculated during the execution of the algorithm and requires minimal post processing upon completion. A full population of circles approximating to the polygon boundary was generated from the medial axis. By definition, these circles pass through at least three original polygon points; these points are stored with the circle that passes through them. This population was reduced (or filtered) according to variables of minimum circle threshold (indicative of permitted error), degree of overlap allowed (which influences the process of filtering) and subsequent introduction of zero-radius circles, defined by various thresholds of the Douglas-Peucker line generalisation algorithm. This population is stored, ready for approximate reconstruction of the original polygon through a combination of circle arcs and dual circle tangents.

2.3 Filtering

Having calculated the full population of circles from the polygon boundary, four stages of filtering are needed to remove unwanted circles. Firstly, those circles with centres within the specified error threshold are removed (Figure 2 illustrates this). The threshold is represented by an epsilon band of spatial uncertainty, as defined by Perkal (1956). To use the terminology of the epsilon band; the minimum circle threshold is equivalent to ε, which is the maximum error distance from the recorded line (as this limits on either side of the original line, the full width of the band is 2ε).

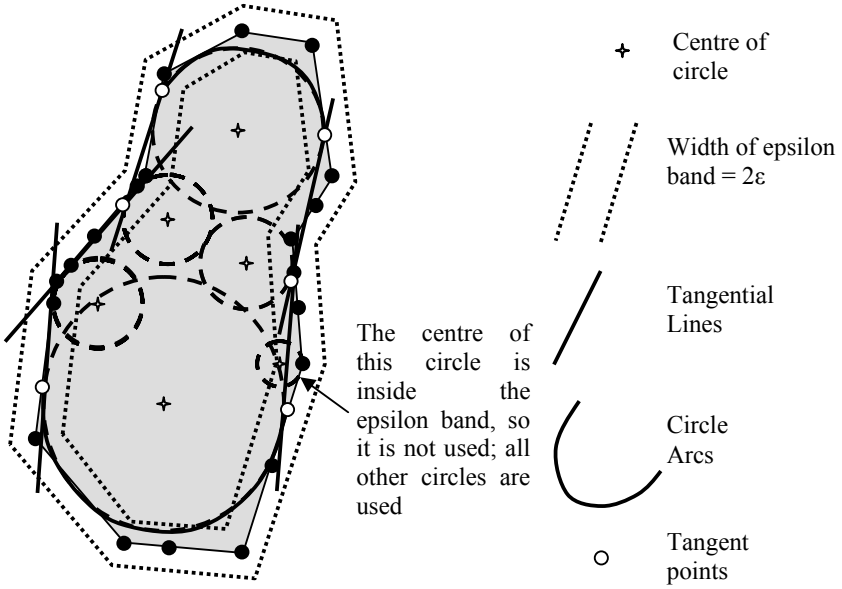

Figure 2. Reconstructing the polygon using tangents and circle arcs

The second stage of filtering involves removing circles that exceed a specified degree of overlap with other circles. A pre-process to this is ordering the circles by ascending radius, keeping the smaller circles so that detail is retained. The third filtering process removes circles whose arcs fall outside the error threshold. In other words, if the length of circle radius (drawn normal to the polygon edge) that is outside of the polygon exceeds the threshold, then that circle is removed. The final stage of filtering removes circles that replace less than two consecutive points on the original polygon. 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.

2.4 Reconstruction of the polygon from stored circles

To effect a tighter fit of the original polygon by circles, two special case circles can be stored. 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, it makes sense to be able to process the boundary from the exterior of the polygon as well. This is accomplished by external circles, stored as (x, y, -r) or with negative radii. The other special case is that of zero-radius circles (x, y, 0), which are "important" points extracted by the Douglas-Peucker algorithm. The introduction of these points was deemed necessary, as the existing method of reconstruction was missing important

features in the original polygon (e.g. inlets). As Figure 2 shows, the reconstructed polygon comprises a combination of tangents and circle arcs. The circles are stored in clockwise order in a linked list and are processed two at a time (1st and 2nd, then 2nd and 3rd,…etc.). The line tangential to both circles (and there are four possible scenarios for this: internal-internal, external-external, internal-external, and external-internal) was recorded. 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.

For a fuller description of the entire algorithm, see Moore et al (2003). The following storage results summary can also be found in the same paper.

2.5 Storage results

For storage, the main concern is whether a set of circles can capture the outline of a polygon to an acceptable accuracy level and still form a smaller dataset than the original polygon data. Two sub-issues were discussed: the role of overlapping circles and the benefits in adding external circles to the conventional internal circle population (Moore et al, 2003).

The circle-packing algorithm was tested on a polygon of the South Pacific island of Rarotonga (Figure 3). This is an example of a relatively smooth coastline and a favourable initial test with which to test the parameters of this algorithm. The map used for this study has 2132 points, which means 4264 data values if both x and y are considered. Based on the (x,y,r) triplet used for circles, anything of a value of 1421 circles or below could be considered a saving in storage.

It was found that with a medial axis approach there was a payoff between overlap and accuracy (less overlap meant less accuracy though efficient storage – more overlap enabled more detail in reconstruction but also promoted loop artefacts), and external circles were also found to optimise the polygon line. However, the accurate reconstruction of the polygon from circles was an issue, leaving the conclusion that the technique was of more practical use as a means of generalisation than as a means of reconstruction. The rigidity of the medial axis technique (normally only three points per circle) used has to be a factor in this. More circles were used to generalise the original polygon boundary when the minimum circle threshold = 5, a reflection of the average point separation magnitude (i.e. minimum circle threshold = 1 is too small for most circles at this level to be of any use in this method of generalisation). Also, it is significant that the circles used for this parameter level "absorbed" the greatest proportion of "important" points (the latter being defined by Douglas-Peucker), meaning that the circles were actually finding a lot of these key points. Other issues exposed in this study included the loss of explicit boundary coordinates.

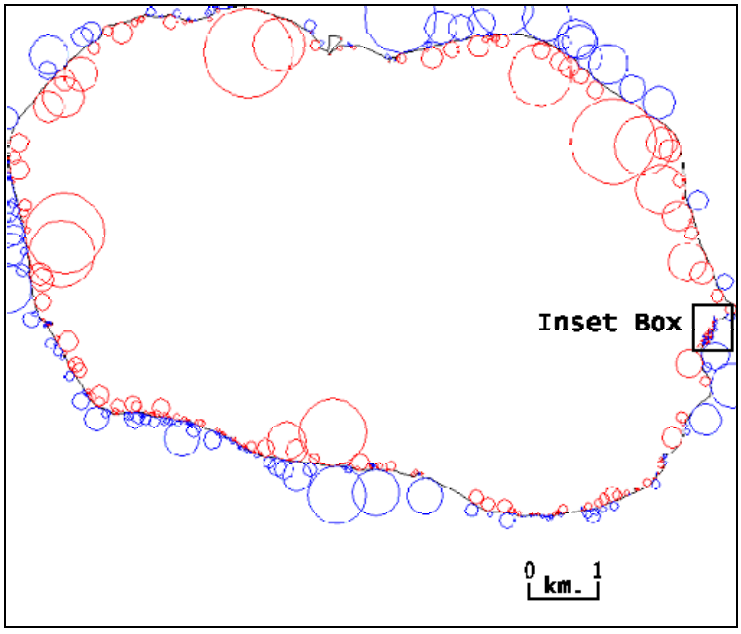

Figure 3. Reconstructed outline of Rarotonga, with internal and external circles (Moore et al, 2003).

2.6 Areal Error Measurement methods

The analysis of error could be taken further, through measuring the sliver area contained between the original polygon and the one constructed from tangents and arcs. A similar method to this (visual error) is used by Alani et al (2001) to measure the difference between polygon boundaries and approximations generated by Voronoi polygon edges (the Voronoi diagram is calculated for point place locations contained within the original polygon boundary). This is a graphic measure, assessing parity of shape. It is given by the formula

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

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). The sum of the two is called the symmetric difference.

A simpler and commonly-used measure is the total areal error, which is given by

$$(\text{Approx area} - \text{actual area})/\text{actual area}$$

Both methods will be used in this research, with amendments being made to the visual error method (see Figure 4). As the approximated boundary from the circle process includes circle arcs as well as tangent points, the arcs have to be discretised into points themselves. This involved the splitting of the arc circumference into smaller meaningful units, the start and end coordinates of which would be the points to be stored. It was decided that the minimum separation between consecutive coordinates (1.96m) would be the unit circumference for arc splitting.

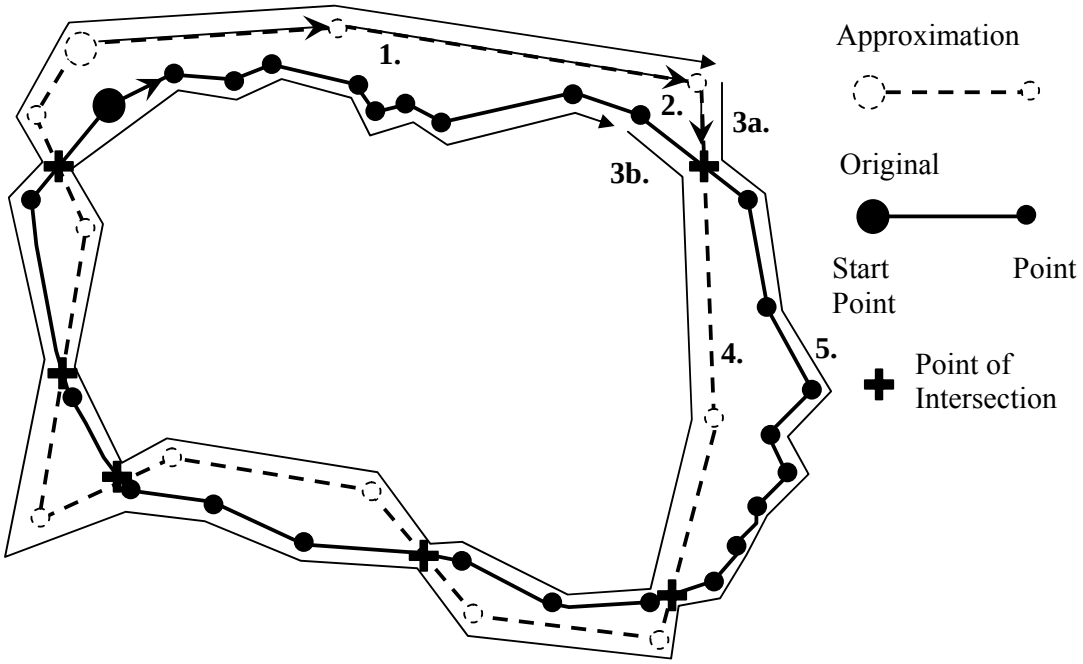

Figure 4. Finding the inner and outer polygons from the original and its approximation. Numbers are referred to in the text. The areal difference between the inner and outer polygons (denoted by the long arrows) is used to calculate the visual error

The process is as follows:

- Input is two polygons, the original and its approximation
- Use a point-in-polygon algorithm to ascertain which polygon is the outer, which is the inner, at the start point

- Starting at the first point pair for the outer polygon, check for intersection with the inner polygon (1). No storing of points occurs at this stage.

- Once intersection occurs, record the point preceding the point of intersection for both the inner and the outer polygons (2). Record the first point of intersection coordinates as the second point in both cases (3a and 3b).

- Process for the inner polygon, again checking for intersection, and storing points as the algorithm progresses along the boundary (4 - 'advance').

- Once intersection occurs, record the point of intersection coordinates, then fill the outer polygon with the outer coordinates since the previous point of intersection (5 – 'catch-up'). Record the point of intersection coordinates for the outer polygon.

- For each subsequent intersection, repeat, switching advance and catch-up processing roles for the inner and outer polygons until the first (and last) point is reached (this is processed as for a point of intersection, with 'advance' and 'catch-up')

- Output is the inner and outer polygon. The symmetric difference is the difference between the areas of the two, and this is divided by the area of the original polygon to give the visual error.

Two preprocesses are necessary to address the loops inherent in some of the approximations, and recurving. To address the former, the polygon was reordered so that line self-intersection did not exist. For instance, assuming a clockwise order, once a self-intersection had been detected, points between the two recorded instances of that intersection are copied into a new array in anticlockwise order. Another special case concerns either the original polygon, or the approximation (or both) recurving on itself without self-intersection, combining with the other polygon to leave an island of unoccupied space, which is erroneously counted as visual error.

3.0 RESULTS

Like the storage tests reported on in section 2.5, the circle-packing algorithm will be tested on a polygon of Rarotonga on the basis of areal error and visual error. The values for the three variables were altered in the same way as in Moore et al (2003). The minimum circle threshold values were tested for values of 1, 5, 10, 15 and 25 metres radius. The amount of permitted overlap was tested for 0%, 10%, 40%, 60% and 90%. Finally, the Douglas-Peucker threshold was altered for values of 1, 5, 10, 15 and 25 metres, and no Douglas-Peucker used. This made for 150 unique combinations. For this presentation, the results will be displayed, discussed and compared with the results of the storage tests.

ACKNOWLEDGEMENTS

This research is supported by a University of Otago research grant. We would like to thank Darrin Drumm for providing the Rarotonga coastline data.

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.

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.

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

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

Jones, C B. (1997). *Geographical information systems and computer cartography*. London: Longman

Jones, C B and Abraham, I M. (1987). Line Generalization in a global cartographic database. *Cartographica*, 24, 3, 32-45.

Moore, A B. (2002). The circle tree – a hierarchical structure for efficient storage, access and multi-scale representation of spatial data. In: P A Whigham (ed), *Proceedings of the 14th Annual Colloquium of the Spatial Information Research Centre, Victoria University, Wellington, New Zealand.* Dunedin, NZ: SIRC, University of Otago, pp.149-156.

Moore, A, Mason, C, Whigham, P and Thompson-Fawcett, M. (2003). The Use of the Circle Tree for the Efficient Storage of Polygons. *Proceedings of GeoComputation 2003,* University of Southampton, UK, CD-ROM.

Odgaard, A and Nielsen, B K. (2003). *A visual implementation of Fortune's Voronoi algorithm.* http://www.diku.dk/hjemmesider/studerende/duff/Fortune. Accessed: 14th July, 2003.

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

Pham, T-L, Schneider, G and Goose, S. (2000) Exploiting Location-Based Composite Devices to Support and Facilitate Situated Ubiquitous Computing. In: P Thomas and H W Gellersen (eds.) *Handheld and Ubiquitous Computing*, Berlin: Springer. pp. 143-156.

Rigaux P, Scholl M and Voisard A. (2002). *Spatial Databases with Application to GIS.* Morgan Kauffmann.

van Oosterom, P. (1993). *Reactive Data Structures for Geographic Information Systems*. Oxford: Oxford University Press.

Wu, J and Amaratunga, K. (2003). Wavelet triangulated irregular networks. *International Journal of Geographical Information Science,* 17, 3, 173-289.