The circle tree – a hierarchical structure for efficient storage, access and multi-scale representation of spatial data

Antoni Moore

Spatial Information Research Centre
University of Otago. Dunedin, New Zealand
Phone: +64 3 479-8138 Fax: +64 3 479-8311
Email: amoore@infoscience.otago.ac.nz

Presented at SIRC 2002 – The 14th Annual Colloquium of the Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
December 3-5th 2002

ABSTRACT

A hierarchical tree structure, the circle tree, is proposed for the efficient storage, access and multi-scale representation of polygonal spatial data. The tree is built by optimally and recursively placing circles of decreasing size to fill a polygon, achieving a near-space filling effect. This paper establishes the precedent for the circle tree in the computer science (e.g. use of volume-filling spheres for quick 3D collision detection) and GIS literature (e.g. the sphere tree as an alternative to the R-tree indexing method) and explains how its development may build on some of these ideas. A project outline describing the comparison of a deterministically derived solution with one gained through evolutionary computation will then be given. Finally, a brief assessment of how the circle tree may facilitate efficient storage and access as well as the representation of a spatial object at multiple scales (and in doing so performing the cartographic processes of reduction and elimination) will be made.

***Keywords and phrases*:** indexing, space filling, cartographic generalisation, sphere tree

1.0 INTRODUCTION

There is a huge amount of data in geography and many other disciplines. This resource will grow exponentially as the means to collect spatial data as well as storing and analysing data become even more ubiquitous. Consequently there is a need to find efficient ways of navigating through this massive geographic dataset.
The project described in this paper aims to develop a novel tree structure, the circle tree, which irregularly partitions space into circles. Two approaches will be applied to build this hierarchical spatial data structure, comparing a tree-based algorithm and evolutionary models to address the data problem. Hierarchical data structures (the most common of which is the quadtree) are a group of methods that recursively subdivide space and can represent spatial phenomena at a variety of geographic scales with successively larger scales represented on successively lower tiers of the hierarchy. They have been used in GIScience as an efficient means of storing spatial data, as a fast way of retrieving that data and as a multiscale representation. They can be space filling (as with the quadtree) or act as an index to a specific and discrete portion of space, with no assumption that space has to be filled (R-tree).

In the proposed project the flexibility of the circle tree algorithm will be demonstrated by applying it

(a) as an efficient data storage mechanism on 1:50000 topographic data
(b) as a fast data access and retrieval means, tested on a very large dataset. The 2001 Census data will meet this requirement, and will also enable an analysis relating to the value of the methods from the view of an end user (urban geography).
(c) as an effective cartographic generalisation technique (in this case generalisation will involve the reduction in complexity of a spatial object), with multiscale representation that retains the essential

shape of a spatial object (this is especially relevant with the demand for rapid display of maps on the WWW). The 1:50000 topographic data set will be the basis for this application.

The effectiveness of the circle tree algorithm (an example of a predefined model) will be compared with models created through a stochastic search algorithm based on evolutionary computation. Evolutionary techniques will be utilized to create a hierarchical model which will be subsequently applied to data access in the ways specified.

The remainder of this paper initially establishes the background of the circle tree, identifying related algorithms in the literature. The circle tree algorithm will then be outlined and discussed within the context of a project outline.

2.0 BACKGROUND

2.1 Use of the sphere tree and other trees in computer graphics

Specific to the circle tree, the nearest precedent in a near space filling sense is the 3D sphere trees used in computer graphics to rapidly create dynamic 3D objects to efficiently detect potential collisions (Hubbard, 1996 – Figure 1). In applications such as virtual reality, a real time processing capability of modelled objects is essential. To maintain such a processing rate, some accuracy is sacrificed to increase speed. Coarse spheres are initially used to approximately render an object, being incrementally refined until a time limit is reached (this time limit is enforced to maintain real time processing). This is an example of 'time-critical computing'. The relevance of this to the development of the circle tree lies in the task of packing space with spheres (the 3D equivalent of the circle) and the parallels to be drawn between coarsening a 3D object and reducing the complexity of a geographic vector object. Like vector, there is also a requirement for optimal accuracy at any level of detail, as assessment of collision may occur at any time.

The circle tree is an example of a hierarchical spatial data structure, the most common of which is the quadtree. The construction of the quadtree involves the recursive subdivision of non-homogeneous raster into four equal-sized quadrants. Relative to the top root node, which describes the whole raster, the four quadrants are represented by four descendent branches forming the tree – this is true of all non-leaf nodes (see Figure 4a). Leaf nodes occur when a quadrant (at whatever level in the tree) is homogeneous (Worboys, 1995).

Related to spheres or pseudo-spheres, Ottoson and Hauska (2002) describe the ellipsoidal quadtree (EQT), a variation on the quadtree. The EQT facilitates global indexing through a quadtree that has been projected onto a geodesic approximation to the globe. Another global indexing method through a hierarchical data structure is the Quaternary Triangular Mesh (QTM) described by Dutton (1996).

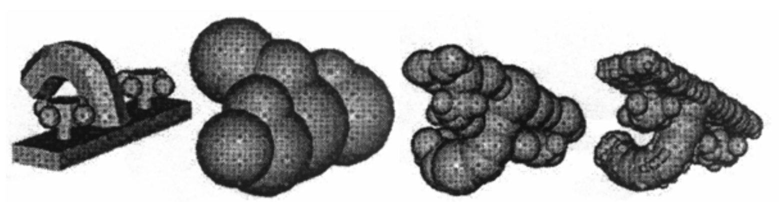

Figure 1: The use of spheres to approximate to 3D objects (from Hubbard, 1996)

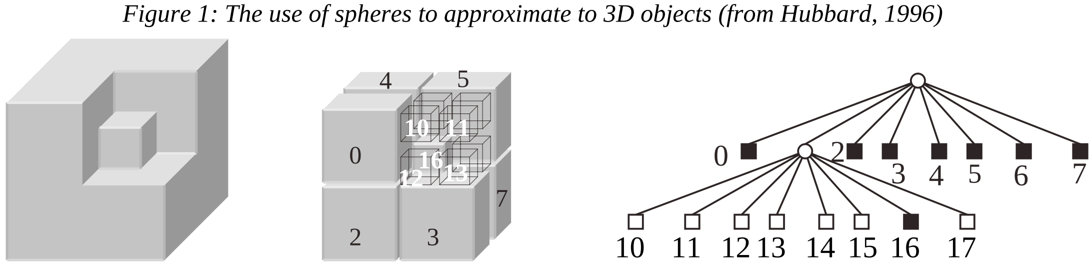

Figure 2: The octree 3D data structure: a) a 3D object; b) recursive decomposition and addressing of the object into octants; c) the corresponding tree (from Worboys, 1995)

The octree (Samet, 1990a, 1990b) is the 3D version of the quadtree, using a $2^n \times 2^n \times 2^n$ array of voxels (volume elements – the 3D equivalent of the pixel) as input for a recursive subdivision into octants (Figure 2b). For each leaf node there are eight descendents (Figure 2c) instead of four (as in the quadtree case). Palmer and Grimsdale

(1995) and Hubbard (1996) both use the octree structure to approximate 3D objects with spheres (the sphere circumscribes any given octant) for quick collision detection, deriving the sphere tree. Hubbard also derives a more accurate representation of a 3D object using the framework of the object itself instead of the octree structure. In Hubbard's structure, the multiple levels of detail are arranged in a hierarchy. At level 0 of the hierarchy is the minimum bounding sphere of an object, which in collision detection acts as a filter to exclude those objects not about to collide.

The Constructive Solid Geometry (CSG) tree (Samet, 1990a, 1990b; Worboys, 1995) constructs 3D objects through a set combination (e.g. union, intersection, difference) and geometric transformation of simple primitive objects (e.g. cubes, cylinders) (see Figure 3). It is primarily of a descriptive nature (i.e. it is not straightforward to derive the underlying properties of an object), but what is of interest for this paper is the possible further decomposition of a primitive instance into half spaces (also stored at lower tiers in the hierarchy). In the case of the circle tree this is analogous to the relationship between a polygon (itself indexed through an R-tree, for example – see section 2.2) and the circles that comprise it.

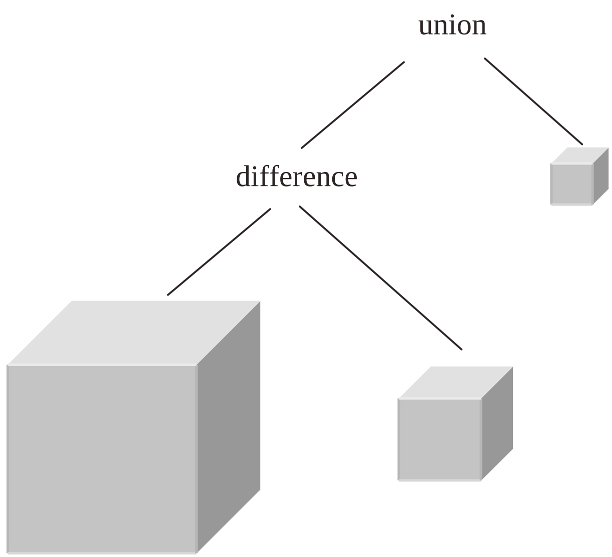

Figure 3: The CSG tree for the 3D object in Figure 2a. The cube is the only primitive instance used (from Worboys, 1995)

2.2 Spatial data indexes

Palmer and Grimsdale (1995) also describe a circular quadtree, the 2D version of the sphere octree described in section 2.1 (Figure 4b). This data structure is the closest to the object of this proposal, except for the pseudo-raster structure and the fact that circles enclose bounding lines as opposed to filling the bounding polygon. This has the distinct advantage that each circle acts as an index to the line it encloses. However this method is not as accurate spatially as letting the edge of the circle approximate to the edge of the polygon.

The sphere tree as defined by van Oosterom (1993) is a variant on the R-tree (described in Rigaux *et al*, 2002), and has been used to store hierarchies of spatial polygons (this is different from the sphere tree described in section 2.1), where for example a 'sphere' (which is actually a circle in the 2D case) bounding a continent would enclose 'spheres' bounding constituent countries (these are minimum bounding spheres - MBS). This is a variant on the minimum bounding box or rectangle (MBB or MBR), which are used by R-trees to index spatial data.

The R-tree (Rigaux *et al,* 2002) may store the index to several MBRs (the precise number is related to the size of a disk block) at a single internal node (these internal nodes also store the smallest rectangle enclosing its descendents). Like its aspatial equivalent, the B-tree, the R-tree uses methods such as node split and merge to avoid overflow or underflow, making for a balanced tree.

The sphere tree needs less space than the R-tree, as it is orientation insensitive. Also, the R-tree (or more correctly the MBR) needs two sets of bounding coordinates, while the MBS just needs the coordinate of the sphere centre and its radius. Other orientation insensitive techniques include some of the family of KD trees, specifically the KD2B-tree (van Oosterom, 1993). Related to the MBS but orientation sensitive are 'Flintstones', which uses two circle arcs that are sized so as to enclose a spatial object in the smallest area possible (cited in van Oosterom, 1993).

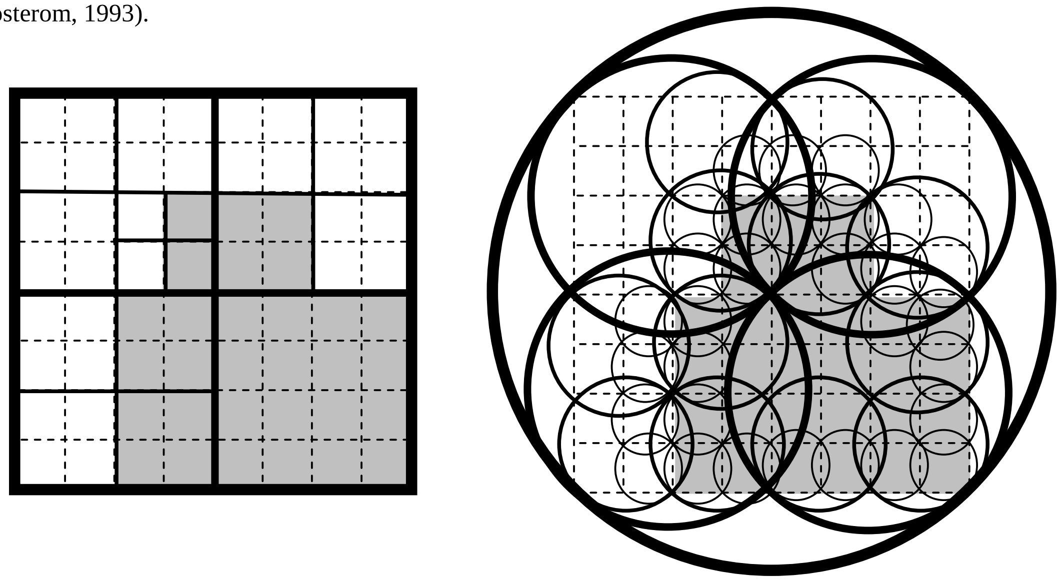

Figure 4: The quadtree (a) and the circular quadtree (b) [adapted from Palmer and Grimsdale, 1995]. The tree structure for both is similar to Figure 2c, except that there are four descendents instead of eight.

2.3 Cartographic solutions

Multiscale representations have been attempted using line generalisation algorithms such as Douglas-Peucker (e.g. Jones and Abraham, 1987) – the use of the circle tree for this purpose is expected to be particularly effective for natural spatial objects which are generally comprised of a series of curves (this is especially true in generalised form). As the Douglas-Peucker algorithm (Douglas and Peucker, 1973) has a recursive form, the first iteration can be used to represent a line at the coarsest level possible without reducing it to a single straight line. With successive iterations, more and more detail is added, and the line at any one of these iterations may be used as a generalised representation. The strength of the technique lies in the ability to store the same line at varying amounts of detail in the same structure.

The Reactive tree (van Oosterom, 1999) is based on the R-tree, specifically addressing the selection element rather than the reduction element of cartographic generalisation. Selection is the exclusion or inclusion of geographic objects on a map with changing scale. Based on the premise that some map objects are more important than others (and this is not necessarily related to size of the object), the Reactive tree promotes such important objects to higher tiers in the hierarchy, making for rapid access.

3.0 PROPOSED METHODOLOGY

The research will be split into four phases.

3.1 Development of the circle tree algorithm

A computer program will be developed, working from the definition of the circle tree algorithm in its current form. Any appropriate refinement of the algorithm will initially take place during this process. The coded algorithm will be tested on a complex spatial data set (coastline from 1:50000 NZ topographic series) to assess efficacy. The algorithm will optimally and recursively fill the area bounded by the coastline with circles until a preset threshold diameter (resolution) is reached. In other words, the algorithm will try and fit a circle inside the polygon so as to maximise the size of that circle.

Lisle (1992) describes an algorithm that takes a distribution of points and tries to fit a circle through them. A least squares approach is used to determine the best circle. Van Oosterom (1993) provides an algorithm to calculate the minimum bounding sphere of a set of spheres or points (e.g. forming a polygon or line). With a dimension of k, select (k+2) spheres from the input set (1) and calculate the MBS (2). If the MBS does not enclose all spheres, mark any spheres not used inside the initial iteration and replace with the farthest sphere(s) (3). Go to step 2 and recalculate the MBS. It is a different sort of problem to calculate the maximal circle enclosed by a polygon. Because of the irregular shape a geographic polygon may take, it does not suffice to use the centroid of that polygon, since it does not indicate exactly where the centre of the maximal circle is and may even lie outside of the polygon. Calculating through internal buffering is an expensive process within a vector framework. However, conversion to a fine enough raster grid and using an appropriate Euclidean distance transform is a relatively time- and resource-effective option (see Moore 2002 for a review of approximate Euclidean distance transforms). Peuquet (1992) uses rasterisation in calculating the Euclidean distance between two vector objects. Instead of calculating the distance outwards from the boundary of the polygon, as is conventional, the distance can be calculated from the cells not occupied by the polygon. Then the cell with the highest value will be the centre of the maximal circle. It should not matter that approximate distances are being used, especially if the output centre coordinates are checked with the original coordinates of the polygon.

Once the maximal circle has been calculated, a similar operation will be performed on the areas not covered by that original circle and so on, filling the polygon with circles of successively smaller size. This task is similar to Apollonian packing, iteratively inserting a smaller circle in the hole formed by three smaller touching circles. However, in the case of the circle tree it is assumed that uncovered space enclosed by three or more circles becomes part of the geographic object. The resulting hierarchy of circles (that have been subject to Apollonian packing) has a self-similar or fractal quality. Herrmann *et al* (1990) describes such packing in relation to the notion of rotating space-filling "ball bearings" of various sizes that facilitate movement between two tectonic plates with minimal friction.

A space-filling effect can be achieved when constructing cartograms. These are maps in which the size of an area (e.g. meshblock) is proportional to an attribute other than geographic area (e.g. population). Dorling (1995) shows some cartograms where geographic shape has been exchanged for the visual clarity of a circle. Inevitably, in the case of population, the proportional circles representing densely-populated areas displace those surrounding it (in a cartogram each circle will own its own area). It is in these areas that near space filling is achieved, though not constrained by real world boundaries. Dorling provides an algorithm to derive this cartogram.

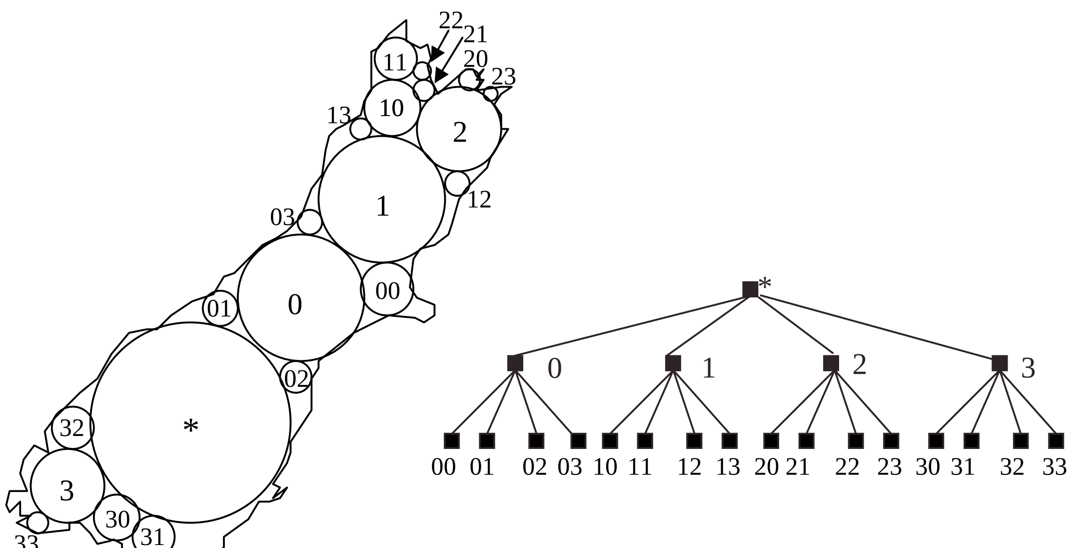

Figure 5: A theoretical circle tree with four descendents: (a) Spatial representation; (b) the tree. No attempt has been made to standardize circle size at a particular level in the hierarchy – this has resulted in an unbalanced tree.

The maximal circle will form the top-level root node of the tree. A certain number n of smaller circles will maximally fill the remaining space and become descendents of the root node. This subdivision of space will proceed recursively until a threshold minimum circle size is reached. These details are subject to change; for example, instead of setting a predefined number n children for each internal node, a variable number of children could be set (however, this would potentially make tree addressing very hard). The latter scenario may be more suited for using the tree as a cartographic selection tool. For such an endeavour it would be good to maintain a similar circle size across a given level of a tree, so that similar-sized polygons are selected or omitted at the same level of scale. This would be facilitated by the flexibility that a variable amount of children would allow.

So the descendents should maximally fill space as a group, but any one descendent circle should be approximately similar in size to any other circle of its level. Figure 5 demonstrates the filling of a polygon with circles and its associated tree. Assumptions may include:

- Together with the node itself, all descendents of a node must form a contiguous group. This will aid in the multiscale representation and also ensure that areas that are close together in space are also close together in the data structure (therefore taking advantage of spatial autocorrelation).
- To ensure optimal coverage of space, any area bounded by a circle node and a portion of the original spatial object must be occupied by at least one descendent (if more than the threshold minimum diameter).

3.2 Other stages

3.2.1 Creation of an evolutionary model

An evolutionary model will be created, so that it performs a similar space-filling role (with circles) when applied to polygons (same data source as above). This is an advance from Krzanowski and Raper's (2001) use of evolutionary algorithms in that it searches a hierarchical tree structure for the best fitting patterns to create the circle tree. This problem has some analogies with bin packing algorithms, however the extension to spatially-structured domains is unique.

While evolutionary algorithms have been used for line generalisation at a particular scale (van Dijk *et al*, 2001), multiscale representation has not been explored in this case. The employment of evolutionary methods to optimally place circles within a polygon so as to fill space is unprecedented, though Krzanowski and Raper (2001) have evolved a location-allocation model for optimal placement of wireless transmitters (the range of which is defined by a circle) so as to maximize coverage.

3.2.2 Comparison of the two methods with established methods

This will be a phase in three parts:

i) Data storage test
Firstly there will be a test for data storage. The circle tree, the evolutionary model and the Douglas- Peucker algorithm (the most popular and effective method of line reduction) will be applied to the same polygonal dataset down to the same level of granularity or resolution. Each resulting dataset will be assessed for size, which is taken to be a measure of data reduction and therefore storage efficiency. The related issue of cartographic representation will be assessed in the third part of this phase.

ii) Data accessibility test
In the second part the two methods and some common tree structures such as the reactive tree will be tested for how they facilitate access to a very large dataset. The 2001 census data set will be used in this case, as it is areal as well as being of the requisite size. This will be effected through measuring the retrieval time for specific subsets of the dataset using the various algorithms.

iii) Multiscale representation test
In the last part of the phase, the two methods will be compared with the Douglas-Peucker algorithm once again, but this time the assessment will be on the basis of how effectively the three algorithms represent a geographical feature at multiple scales, having been generalised. A feature will be chosen at 1:50000 scale, the three methods will be applied to it, and the resulting representations will be compared with the representations of the same feature at smaller scales (e.g. 1:100000) that have been generalised manually by a cartographer. In this way the effectiveness of each method in maintaining the feature's essential spatial character will be judged. This also has implications on what happens to features when maps change scales, which is a very important research problem in GIS.

3.2.3 Critique of the two methods from an end user perspective

As a counterpoint to the assessment in an algorithmic and data issue sense, there will be a critique of the value of the two methods from an end user perspective. The application will be urban geography, using the 2001 census data as the base data set. The critique will primarily assess the social implications of a scale-based rapid access method to subsets of this large dataset. In doing this, the methods will be placed and assessed within a framework in which the object data will be used in an operational sense.

4.0 DISCUSSION

Although there is no close match to the circle tree in the literature, a number of parallels can be drawn. In a 2D sense, the circular quadtree differs in that the circles enclose spatial objects (lines) rather than being used to optimally fill areas. It is when a comparison is made with the 3D sphere tree that a match is made with the mode of algorithm operation (spheres being used to fill volume; circles being used to fill area). Taking the comparison further, the use of spheres represents a coarsening or generalization of a 3D object, which is precisely what is happening when circles are being used to approximate to 2D spatial objects. Ways to improve that approximation have yet to be suggested, but a relaxation of the implicit premise that a circle cannot fall outside of a polygon may be made, as long as the circle boundary is within a threshold distance of the actual boundary of the polygon. The case of a concave form may require a specially coded circle to lie (almost) wholly outside of the object polygon to optimally capture the form.

There are more parallels to be made. Assuming the operation of an R-tree or sphere tree on the polygons as whole entities, there is scope for the development of the circle tree so that it seamlessly becomes an extension of the (R or sphere) tree, but operating at a larger scale (within the polygon). A similar effect has been observed with the CSG-tree: the primitive instances play the part of a polygonal object and the half spaces comprising the primitive instances are the equivalent of circles comprising the geographic object. In doing this, the heuristic shape (MBS) has been used to define the feature shape (circles make up the form of the polygon). This is the key to why the circle tree can facilitate cartographic reduction. However, the gaps between circles may introduce some unrealistic artefacts, showing non-existent “inlets” on the cartographic representation (to get an idea of this, trace around the outer arcs of the outer circles in Figure 5a and compare the result with the real coastline). A possible solution may lie in linking two adjacent outer circles by a line tangential to both outer arcs (checking that no prominent features are being eliminated).

Also, an optimal tree build (see section 3.1 for discussion on this) arranges circles of similar size at the same tier in the hierarchy, allowing a standardized cartographic selection process to take place. This task has conventionally been effected by the Reactive tree. The potential accommodation of two cartographic processes by the same tree is reckoned to be a key property.

It is anticipated that the circle tree will work best with natural features (especially any curvy, sinuous rivers), which tend to be irregular and approximately curved. In addition, certain natural forms such as coastlines (Mandelbrot, 1967) and man-induced forms such as city boundaries (Batty and Longley, 1994) have been shown to possess a fractal or self-similar nature. Hermann *et al* (1990) has said the same about space filling circles. With these two fractal spatial phenomena operating side-by-side, there is a real opportunity if the two fractal entities can be made to work in a complimentary way, rather than at odds with each other. The potential for conflict is suggested by the implied mismatch between circle and original coordinates. With coastlines, which have an inherently rough form (which can be modelled by fractals), the circle tree is expected to have a smoothing effect. Man-made features with straight sides (this describes most man-made features) are not expected to be modelled to any great accuracy by circles.

5.0 CONCLUSIONS

This paper outlines a proposed project that aims to investigate and report on:

- the development and application of the circle tree for storing, accessing and multiscale representation of two large and complex spatial data sets
- using evolutionary techniques, the creation of a hierarchical model to be applied to the same domain
- comparison of a predefined circle tree model and the evolved circle tree model
- critique of the methods from the view of a possible end user, indicating value in an operational sense

From this account, there are several paths of innovation, in the development and flexible use of the circle tree and the use of evolutionary methods in a tiered sense. Added to this is the value of an investigation into the use of the methods from an operational perspective (census-based urban geography).

ACKNOWLEDGEMENTS

I would like to thank Peter Whigham (Information Science) and Michelle Thompson-Fawcett (Geography) of the University of Otago for their continuing input into this proposed project, which has been funded by a University of Otago Research Grant (UORG) in the Academic Year 2003.

REFERENCES

Batty, M and Longley, P. 1994. Fractal Cities. Academic Press.

Dorling, D. 1995. Visualizing changing social structure from a census. *Environment and Planning A*, 27, 353-378.

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. 1996. Improving locational specificity of map data – a multi-resolution, metadata-driven approach and notation. *International Journal of Geographical Information Systems*, 10, 3, 253-268.

Herrmann, H J, Mantica, G and Bessis, D. 1990. Space-Filling Bearings. *Physical Review Letters*, 65, 26, 3223-3226.

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

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

Krzanowski, R and Raper, J. 2001. Spatial Evolutionary Modelling. Part II: Algorithms and Models, 61-124. Oxford University Press, Oxford.

Lisle, R J. 1992. Least Squares Best-Fit Circles (with applications to Mohr's diagram). *Mathematical Geology*, 24, 5, 455-461.

Mandelbrot, B B. 1967. How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science* 156, 636-638.

Moore, A B, 2002. The case for approximate Distance Transforms, this volume.

Ottoson, P and Hauska, H. 2002. Ellipsoidal quadtrees for indexing of global geographical data. *International Journal of Geographical Information Science*, 16,3, 213-226.

Palmer, I J and Grimsdale, R L. 1995. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14, 2, 105-116.

Peuquet, D. 1992. An Algorithm for Calculating Minimum Euclidean Distance between two Geographic Features. *Computers and Geosciences*, 18, 8, 989-1001.

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

Samet, H. 1990a. Applications of Spatial Data Structures. Addison Wesley.

Samet, H. 1990b. The Design and Analysis of Spatial Data Structures. Addison Wesley.

van Dijk, S, Rhierens, D and de Berg, M. 2001. Designing Genetic Algorithms to Solve GIS Problems. In: Krzanowski, R and Raper, J, *Spatial Evolutionary Modelling*, 158-179. Oxford University Press, Oxford.

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

van Oosterom, P. 1999. Spatial Access Methods. in Longley, P., Goodchild, M., Maguire, D and Rhind, D. (eds) *Geographical Information Systems* Vol.1 385-400. John Wiley.

Worboys M. 1995. GIS – A Computing Perspective. Taylor and Francis.