DIGITAL MAP GENERALIZATION USING A HIERARCHICAL COORDINATE SYSTEM

Geoffrey Dutton
Department of Geography
University of Zürich
Winterthurerstrasse 190
8057 Zürich Switzerland

dutton@geo.unizh.ch

Abstract

The use of hierarchical coordinate systems in geographic information systems (GIS) is a relatively unexplored area, particularly with respect to cartographic generalization techniques. This paper describes a hybrid geospatial data model that enriches vector- topological descriptions of map features by quadtree encoding of vertex locations. It also summarizes methods to encode, analyze, filter and decode vector map data for display at scales smaller than those at which they were captured. Geometric and combinatorial computations are performed either on absolute quadtree addresses, on a world projection or directly on the sphere. The software platform presently only processes one feature class at a time, but is intended to handle more, whether stored as overlaid coverages or as independent or linked objects. Map generalization computations are localized using hierarchical hexagonal and triangular cells called Attractors. This "space-primary" approach to map generalization does not depend upon a hierarchical feature classification scheme, but the two perspectives are related and could be united. This paper describes (1) the quaternary triangular mesh (QTM) hierarchical location encoding scheme; (2) modeling of cartographic features; (3) some new generalization algorithms and conflict detection techniques; and (4) potential benefits of applying this approach across feature classes.

Hierarchical Map Generalization

Thematic Hierarchies. Approaches to hierarchical map generalization fall in two main categories. As used by some researchers (Molenaar 1996, Richardson 1994), the term refers to techniques for merging or eliminating map objects based on hierarchical feature classification. Constraints to minimum size and adjacency are normally used to eliminate or merge map features represented as polygons. The examples most often given tend to involve generalizing land use maps, which assign nominal codes to polygons at several levels of specificity, such as rural - agricultural - cropland - cornfield, or urban - industrial - transportation - railyard. Merging adjacent areas having the same use code (or removing inclusions smaller than a certain size) results in a simplified map, although the amount of line detail of the remaining polygons is not decreased accordingly. This paper does not address such possibilities, but thematic object hierarchies could potentially be used to compute semantic priority constraints for negotiating conflicts among multiple map features.

Geometric Hierarchies. Most geometric approaches to hierarchical generalization work by progressively eliminating vertices describing the importance of vertices along polylines and polygons in a consistent manner (Cromley, 1991). Doing this insures that vertices selected to define features at larger tolerances (i.e., lower resolution or bandwidth) are retained when tolerance is reduced, as additional vertices are selected which are likewise retained at yet-smaller tolerances. This also means that once a vertex is removed for display purposes, it will not reappear at smaller scales. Non-hierarchical generalization methods do not inherently include previously-selected points (nor do they always exclude previously-eliminated ones) when tolerance is changed, and this can sometimes lead to inconsistent representations, especially when zooming in and out interactively. van Osteroom (1993) and van Osteroom and Schenkelaars (1995) describe a hierarchical implementation of the widely-used Douglas line simplification algorithm (Douglas and Peucker 1973) that constructs a tree of vertices that can be repeatedly accessed to retrieve line detail at specific scales, achieving a *multi-resolution representation* (versus *multiple representations*). We take a different approach to hierarchical map generalization, using hierarchical coordinates and on-the-fly vertex selection. As in hierarchical data structures such as strip trees (Ballard 1981), various levels of detail are encoded, but being quadtree-based, it is a hierarchical partitioning of space rather than of phenomena. The space is a polyhedral approximation of a planet, and the phenomena are represented as strings of quadtree leaf addresses, each representing a two-dimensional geographic coordinate pair. This strategy combines the scale-sensitivity and indexing capabilities of quadtrees with the flexibility and rigor of vector-topological data models, as well as being able to handle data encoded as isolated features.

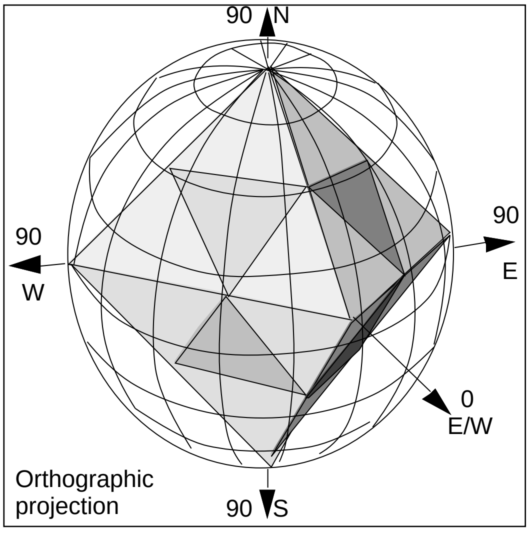

Fig. 1: An Octahedron Embedded in the Earth

Quaternary Triangular Mesh (QTM) is a global spatial indexing scheme and a hierarchical coordinate system proposed by Dutton (1989) for managing GIS positional data quality. A similar model was developed around the same time by Fekete (1990) for indexing and browsing remote sensing imagery. Various uses for QTM and related encodings have been explored by various researchers (global spatial indexing and visualization: Goodchild and Shirin 1992, Otoo and Zhu 1993; Terrain data compression: Lugo and Clarke 1995; positional data quality: Dutton 1992, 1996). A hierarchical approach to map generalization using QTM was proposed by Dutton and Buttenfield (1993), but not implemented until recently (Dutton 1996a). Work reported here further explores this line of investigation.

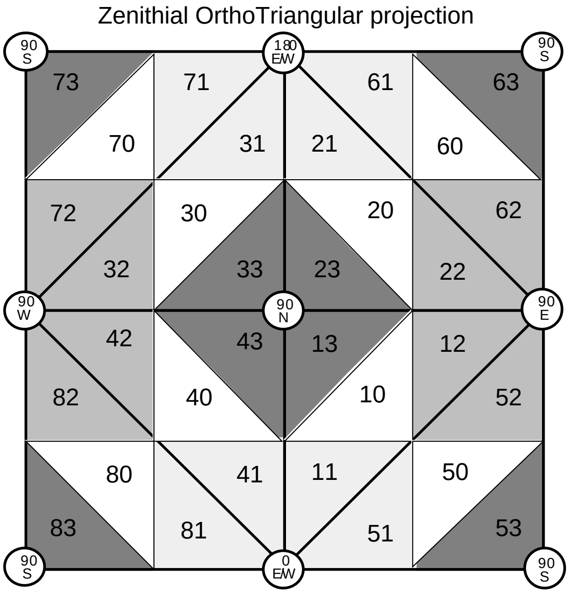

Fig. 2: QTM Octant and level 1 facet numbering

QTM Hierarchical Coordinates. To carve a planet's surface into a quaternary triangular mesh, a virtual octahedron is embedded in the earth, its vertices aligned to cardinal points (see fig. 1). The octa faces form the roots of a forest of triangular quadtrees; these eight triangles recursively bulge into four children. Child *facets* are identified by integer codes having up to 30 quaternary (base 4) digits, enabling locations on the planet as small as 2 cm^2 to be uniquely identified (spatially indexed).

The addressing scheme, while planetary in scope, is capable of handling regions smaller than land parcels, but processing time increases as precision goes up. A binary representation for QTM identifiers has been proposed (Dutton 1996) that uses 64 bits per address, usually with a number of bits left over that may be used to store other properties of points beside location. Geographic points in latitude and longitude are converted to QTM codes via an octahedral projection: more accurate points get longer addresses. The algorithm used to encode and decode point data can be implemented either recursively or iteratively. Figure 2 shows the octahedral projection and the QTM numbering scheme.

Map Encoding. Because different GIS vendors and their applications organize geographic phenomena using different schemata, QTM data processors should make as few assumptions as possible regarding data models. Therefore, all the processing techniques described below handle data at the "feature primitive" level: strings of geographic coordinates, with or without explicit topological relations. Vector map data are modeled as *features* via the *primitive elements* (point sets, polylines, polygons) that comprise them; each primitive is a set (or list) of coordinates, and features consist of lists of primitives. As a given primitive may participate in more than one feature (such as a river that serves as a property boundary), each primitive identifies the features that use it. A master catalog identifies and summarizes the locations (bounding rectangles and enclosing QTM facets) and the logical relationships of all elements. Any collection of features may be designated as a *feature set*, which can model logical, topological, positional or thematic dependencies. This storage and access architecture is diagrammed in figure 3. Note the derivation of QTM addresses and Attractor addresses from the primitives' coordinates (assumed to be latitude/longitude).

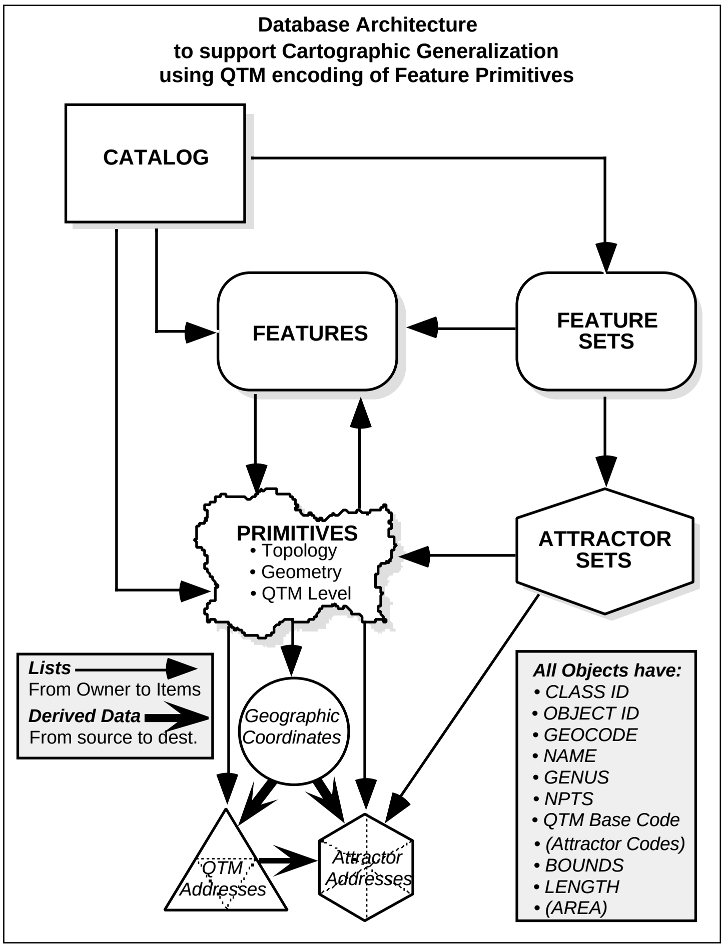

Fig. 3: Basic QTM vector feature data model

Generalizing Spatial Primitives

In preparing digital maps for generalization, all coordinates are converted to hierarchical QTM addresses, at a level of precision appropriate to their positional accuracy. This pre-processing is diagrammed in figure 4. Should positional data quality be unknown, it may be estimated by statistical analysis of QTM-filtered coordinates (Dutton and Buttenfield 1993); for medium-scale maps QTM addresses are from 15 to 25 digits long). Should features or regions be of differing accuracy, this variability can be modeled throughout processing by varying the length of QTM addresses. QTM facets group themselves into hexagonal regions, which figures 3 and 4 indicate as *attractors*. These serve as "buckets" to collect vertices and identify primitives that are likely to conflict at specific map scales. Attractors are hierarchical, but in a more complicated way than QTM facets are. They are implemented as transient, dynamic data structures (objects or lists) for conflict detection purposes only. Attractors facilitate spatial search because they contain sets of facets that are *cousins* (rather than *siblings*).

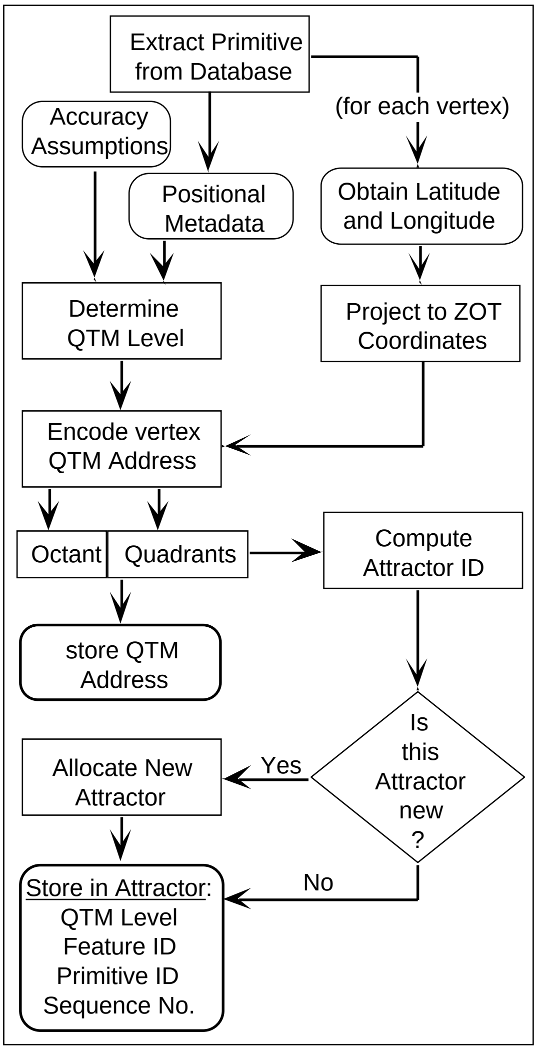

Fig. 4: Computing QTM Hierarchical Coordinates for vertices of a primitive

Whatever its source may be, the positional accuracy of map primitives needs to be expressed in linear units of ground resolution (e.g., cm) in order for QTM encoding to occur. Each QTM level of detail has a characteristic (but slightly variable) linear resolution — the edge lengths of facets making up the level. QTM encoding halts when mean resolution would drop below the linear accuracy for a primitive. To give some examples, the resolution of QTM level 17 data (76 m) is comparable to that of Landsat scenes. Level 20 resolution is about the size of a SPOT pixels (10 m), level 24 (60 cm) can resolve objects big as doormats, and level 30 (2 cm) can locate fence posts.

When digital map data is believed to be oversampled, the encoding process shown in figure 4 may include an extra step: the QTM IDs of successive vertices are compared, and duplicates are weeded out. This simple operation, when applied to attractors, is the core of a set of QTM generalization techniques.

Figure 5 shows the general structure of attractors with four levels superimposed. The triangular areas between the hexagons are also attractors; these contain single QTM facets rather than sets of six, and assist in relating the three hexagonal ones that surround them (which touch only at vertices). An actual set of attractors computed for a polygonal feature (part of the Swiss canton of Schaffhausen) is shown in figure 6 in equi-rectangular projection. Their skewed appearance reflects the shape of QTM facets in that part of the world (47.5° N, 8.5° E; attractors form perfect hexagons only at the eight QTM octant centers). Identifiers for attractors are arbitrary; currently one of the QTM facets within each attractor is selected to name it. Hence both QTM IDs and AIDs have addresses of the form OQQQ...QQQ, where O is octal and Q represents quaternary digits.

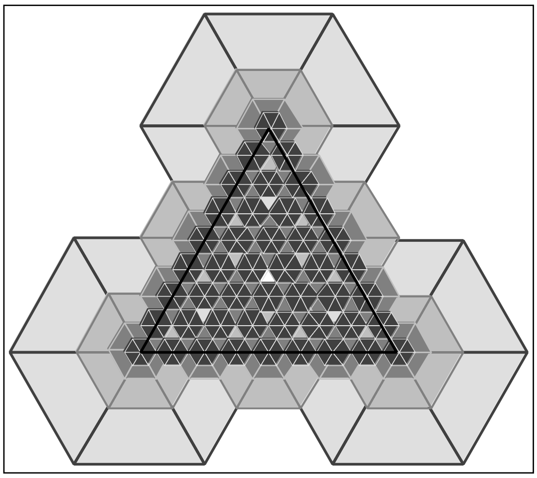

Fig 5: Four levels of attractors for a QTM facet

QTM Detail Filtering. To remove line detail when reducing a feature's scale, an appropriate level of attractor is computed for every vertex along the polyline(s) that form it, as fig. 6 shows. The basic filtering operation for individual primitives consists of scanning the sequence of vertices to determine which adjacent ones share a given attractor (non-adjacent points can also be compared). One vertex is then selected to represent all the vertices that fall in each attractor. This can be handled in various ways.

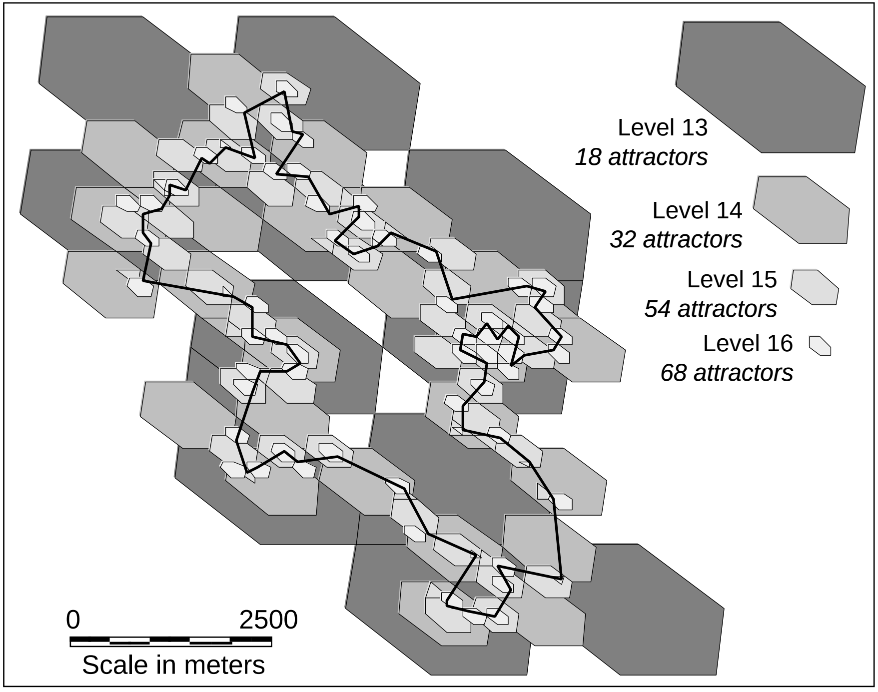

Fig. 6: Attractors occupied by vertices of a polygon having 73 points

Internally, each primitive has two string arrays allocated for it, one holding the QTM IDs of vertices, the other holding their Attractor IDs (AIDs). The array addresses are passed to a filtering function, which scans the AIDs for runs of successive duplicates. When a run is detected, a single vertex is selected from among that set of vertices to represent all of them. Which vertex is selected can make a difference in the appearance of the generalized primitive; a number of different criteria may be applied in making this choice:

1. Primitive endpoints (topological nodes) are always retained
2. Longer (more precise) QTM IDs, if any, are preferred over shorter ones
3. Vertex-specific positional metadata, if any, can be consulted
4. Longer line segments may be preferred over shorter ones
5. Sharper vertex angles may be preferred over less acute ones
6. Vertices nearest the middle of their runs may be preferred

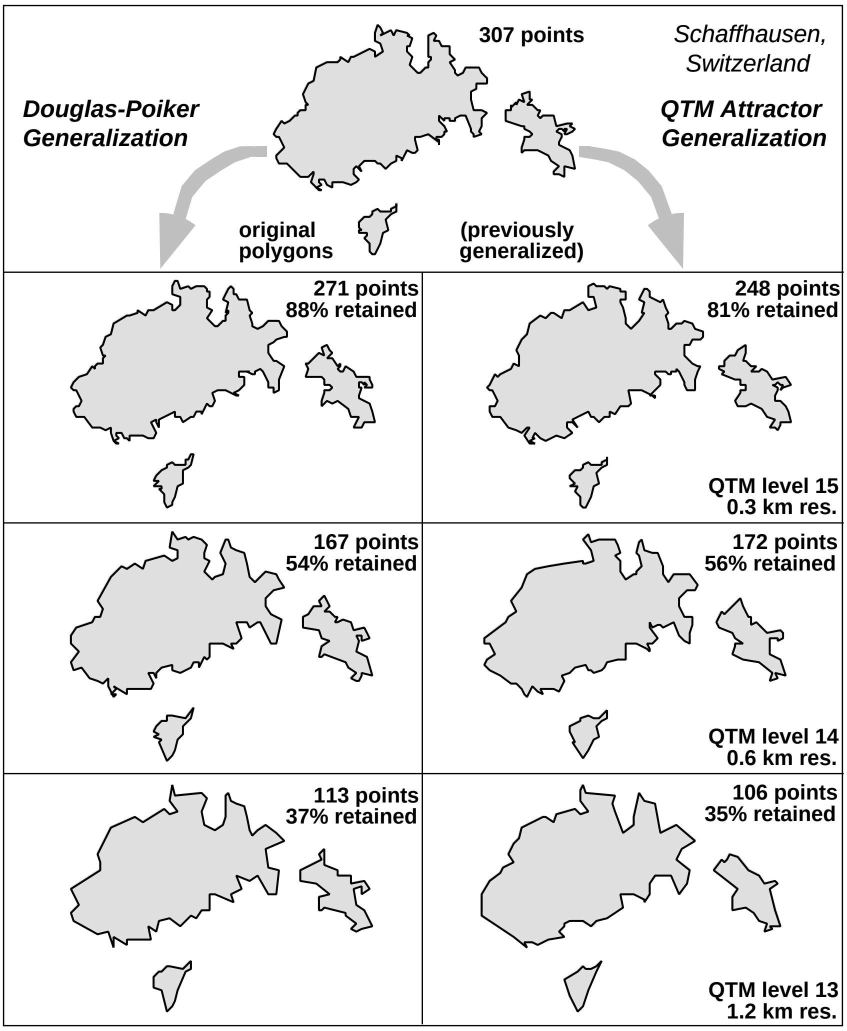

Fig. 7: Comparison of output of QTM "Median Attractor" and Douglas-Peucker Line Generalization Methods

Geometric criteria, such as items 3 and 4 in the above list, cannot be directly evaluated by functions that handle arrays of ID strings. Such evidence must be gathered when compiling QTM IDs; the results of geometric analyses can be coded into each ID in the form of *qualifiers*, as described by Dutton (1996). When metadata is unavailable, the default decision is to select the median vertex from a run, or if there are a pair of them, the one that has the lexigraphically largest QTM address. This is somewhat arbitrary, but yields repeatable results.

Preliminary Results. The "median attractor" method of vertex elimination just described has been tested on individual map features with quite reasonable results. Figure 7 summarizes a multi-scale generalization of a Swiss canton, comparing these results to the Douglas-Peucker algorithm. The boundary data was originally digitized by the Swiss federal mapping agency in official national grid coordinates. Four files having differing levels of detail were derived from that source data using the Douglas algorithm, but the tolerance parameters were not documented. All data files were subsequently de-projected to geographic coordinates; the most detailed of these was used as input data for the QTM generalizations illustrated in fig. 7, simply by choosing a QTM hierarchical level at which to output filtered coordinates. Figure 8 displays the same results, but scaled to dimensions at which the feature might appear in printed maps.

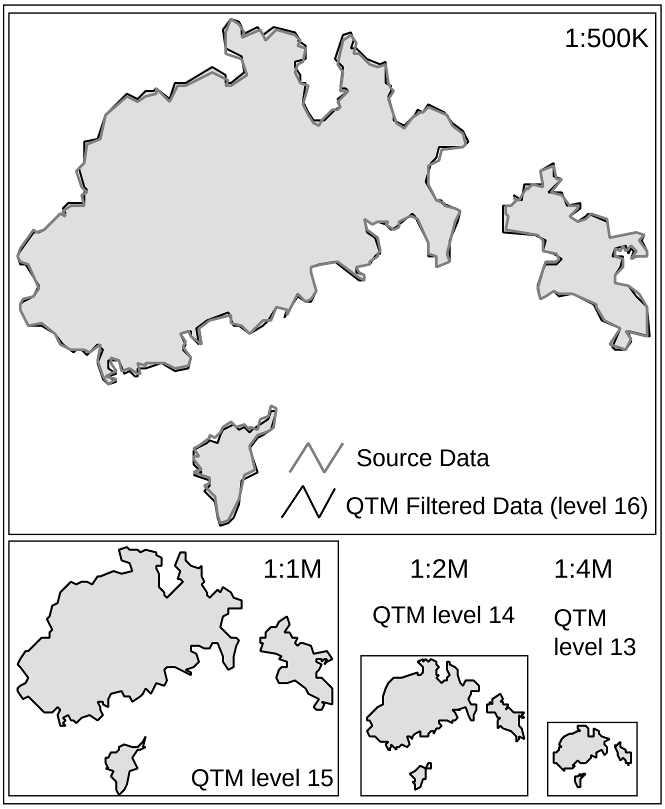

Fig. 8: QTM Generalization Results at Appropriate Scales

Multi-feature Generalization Strategies

The tests described above have dealt only with simple features, and did not explore cases where crowded features compete for map space. Multi-feature (sometimes called "holistic") map generalization is, of course, a much more difficult problem, one which seems to require a diverse mix of strategies, as recent literature evidences (Müller et al 1995; Ruas and Plazenet 1996; Ware and Jones 1996). Most prior work in this area uses one or more "object-primary" techniques, which the nature of vector-topological GIS data structures makes necessary; proximity relations between spatial objects must be explicitly (and expensively) modeled to detect scale-related conflicts. The alternative is to use "space-primary" methods, which are normally restricted to raster data environments; in them, space is modeled and objects are attributes of locations. Few approaches to vector-based, space-primary map generalization have been developed (see Li and Openshaw 1993 for a rare example), but this perspective may have considerable heuristic value. Populating lists of QTM attractors with pointers to primitives that occupy them is one way to combine space- and object-primary approaches to detect conflict among any number of features. Choosing the size (level) of attractors allows one to cast as fine or coarse a net as one wishes, and QTM's spatial indexing properties can be used to restrict the search space.

It is already possible to export scale-specific, QTM-filtered versions of geodata (i.e., multiple representations) to GIS databases. Eventually, QTM-encoded map data may reside in GIS databases themselves, providing built-in multi-resolution capabilities. To make either approach work, additional processing and decision-making will be necessary to generalize QTM-encoded features for display, as different selections of features (and different purposes and applications) will require continually revisiting regions of interest to make new decisions about how to portray them. How to resolve cartographic conflicts may never be easy to decide, but at least we will know what they are and where to find them.

References

Ballard DH 1981. Strip Trees: A hierarchical representation of curves. CACM 24(5): 310-21

Cromley RG 1991. Hierarchical methods of line simplification. Cartography and Geographic Information Systems 18(2): 125-31

Douglas DH, Peucker TK 1973 Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10(2): 112-22

Dutton G 1989. Modelling locational uncertainty via hierarchical tessellation. Accuracy of Spatial Databases, M. Goodchild & S. Gopal (eds.). London: Taylor & Francis, 125-140

Dutton G 1992. Handling positional uncertainty in spatial databases. Proc. Spatial Data Handling Symp. 5. Charleston, SC, August 1992, v. 2, 460-469

Dutton G, Buttenfield BP 1993. Scale change via hierarchical coarsening: Cartographic properties of Quaternary Triangular Meshes. Proc. 16th Int. Cartographic Conference. Köln, Germany, May 1993, 847-862

Dutton G 1996. Improving locational specificity of map data: A multi-resolution, metadata-driven approach and notation. Int. J. of GIS. London: Taylor & Francis, 253-268

Dutton G 1996a. Encoding and handling geospatial data with hierarchical triangular meshes. Proc. 7th Symp. on Spatial Data Handling. Delft NL, August 1996, 8B.15-28

Fekete G 1990. Rendering and managing spherical data with sphere quadtrees. Proc. Visualization '90 (First IEEE Conference on Visualization, San Francisco, CA, October 23-26, 1990). Los Alamitos CA: IEEE Computer Society Press

Goodchild MF, Yang Shiren 1992. A hierarchical data structure for global geographic information systems. CVGIP, 54:1, pp. 31-44

Li Z, Openshaw S 1993. A natural principle for the objective generalization of digital maps. CaGIS 20(1): 19-29

Lugo JA, Clarke KC 1995. Implementation of triangulated quadtree sequencing for a global relief data structure. Proc. Auto Carto 12. ACSM/ASPRS, 147-156

Molenaar M 1996. The role of topologic and hierarchical spatial object models in database generalization. In Molenaar M (ed) Methods for the generalization of geo-databases. Publications on Geodesy, Delft, Netherlands Geodetic Commission 43: 13-36

Müller J-C, Lagrange J-P, Weibel R 1995 (eds) GIS and generalization: methodological and practical issues. London, Taylor & Francis

Otoo EJ, Zhu H 1993. Indexing on spherical Surfaces using semi-quadcodes. Advances in Spatial Databases (Proc. 3rd Int. Symp. SSD'93), Singapore, 510-529

Richardson D E 1994. Generalization of spatial and thematic data using inheritance and classification and aggregation hierarchies. In Waugh, TC, Healey RG (eds) Advances in GIS research (Proceedings Sixth International Symposium on Spatial Data Handling): 901-20

Ruas A, Plazanet C 1996 Strategies for automated generalization. In Kraak M J, Molenaar M (eds) Advances in GIS research II (Proceedings 7th Int. Symp. on Spatial Data Handling): pp. 6.1-6.18.

van Oosterom P 1993. Reactive data structures for geographic information systems. Oxford, Oxford University Press

van Oosterom P, Schenkelaars V 1995. The development of a multi-scale GIS. Int. J. of GIS 9(5): 489-508

Ware J M, Jones C B 1996 A spatial model for detecting (and resolving) conflict caused by scale reduction. In Kraak M J, Molenaar M (eds) Advances in GIS research II (Proc. 7th Int.Symp. Spatial Data Handling): 9A.15-26