

Encoding and Handling Geospatial Data with Hierarchical Triangular Meshes

Geoffrey Dutton

Department of Geography
University of Zürich
Winterthurerstrasse 190
CH-8057 Zürich
dutton@geo.unizh.ch

Abstract

Planetary geocoding using polyhedral tessellations are a concise and elegant way to organize both local and global geospatial data that respects and documents locational accuracy. After a brief review of several such spatial referencing systems, topological, computational, and geometric properties of one of them are examined. The particular model described in the remainder of the paper – the *octahedral quaternary triangular mesh* (O-QTM) – is being developed to handle and visualize vector-format geodata in a hierarchical triangulated domain. The second section analyzes the geometric regularity of the model, showing that its facets are relatively similar, having vertices spaced uniformly in latitude and longitude, and areas that vary by less than 42 % from their mean sizes. Section 3 describes some fundamental operations on this structure, including mapping from geographic coordinates into O-QTM addresses and back again, filtering map detail through the triangular hierarchy and associating locations that are close together, but in different branches of the tree structure. The final section outlines and illustrates a recent application of O-QTM to map generalization, using its multi-resolution properties to enable multiple cartographic representations to be built from a single hierarchical geospatial database.

1 Hierarchical Polyhedral Modeling of Planetary Locations

Mapmakers and others have attempted to model the earth as a polyhedron for many years, going back to at least the time of the German artist Albrecht Dürer (1471-1528), whose drawings of polyhedral globes appear to be the first instance of thinking about mapping the planet in this way. In the late 19th and early 20th centuries, a number of cartographers, such as Cahill (Fisher and Miller, 1944; also used by Lugo and Clarke, 1995 in their variant of QTM), reinvented this idea, projecting the land masses of the Earth to various polyhedra, then unfolding their facets into flat, interrupted maps. The best known of these is R. B. Fuller's Dymaxion projection, dating from the early 1940's (Unknown, 1943). Originally based on a cubeoctohedron, the Dymaxion Map was then recast as an icosahedron, oriented to the earth in a way that minimized the division of land areas between its 20 facets. Fuller devised a projection method -- only recently well-enough understood to implement digital algorithms for it (Gray, 1994; also see Snyder, 1992) -- that has remarkably little distortion.

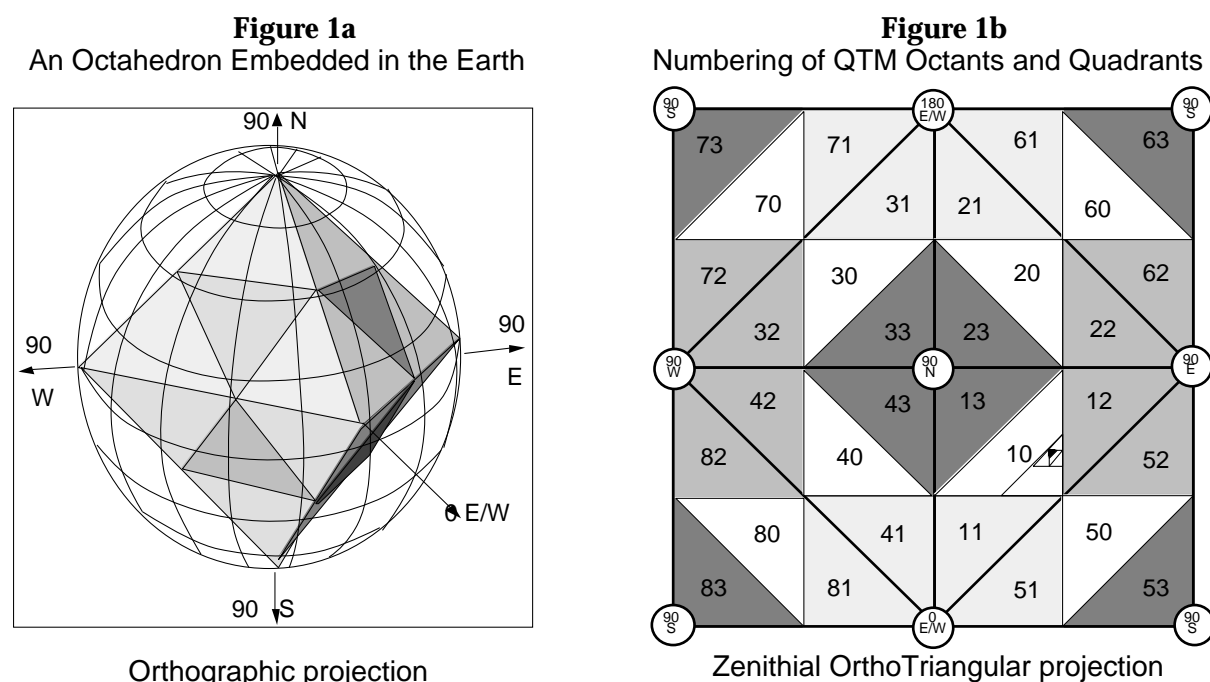
Pre-computer polyhedral projections all were based on platonic or other simple shapes, and were intended as amusements or devices that let a paper globe be unfolded to lie flat. They did not attempt to deal with more complicated cases involving subdivision of polyhedral facets into smaller ones that would more closely fit the figure of the Earth and have more inherent accuracy. Fuller's geodesic domes are examples of how a physical polyhedron can be subdivided to give it greater physical strength per unit weight and span larger areas than any simple polyhedron is capable. Fuller apparently never treated the Dymaxion map in the same manner, probably because he viewed it as a physical structure that was nearly optimal for display of global thematic data, and bringing it to the next level of complexity (either 60 or 80 facets, depending on the method of subdivision) would make the map unwieldy to manufacture and manipulate, and achieve no particular benefit. Now, it's a different world.

The digital revolution in mapping and associated geoprocessing techniques have freed mapmakers from such practical constraints and physical limitations, and over the past 30 years a number of approaches have blossomed for using polyhedra to index and display spatial

data on a world-wide basis for a variety of purposes. Several such models are summarized in order to indicate how they are alike and in what respects their form and properties differ.

1.1 The Octahedral Quaternary Triangular Mesh (O-QTM) Framework

Inspired by Fuller's maps and domes, the author developed a global digital elevation model in 1983 that divided a planet into facets defined by a concentric octahedron and cube. (Dutton, 1984). In 1988 this model was revisited, revised and recast as a tool for spatially indexing planimetric data in a geographic information system (Dutton, 1989). The cube was discarded, but the octahedron remained as a geometric basis that roots a forest of eight quadtrees containing roughly equal triangular quadrants (facets) that approximate a sphere quite closely after only a few subdivisions. Figure 1a depicts the basis of the O-QTM model, an octahedron embedded in a spheroid. Figure 1b illustrates the quadrant numbering scheme, using a map projection that renders every facet at each level as an isosceles right triangle. Use of this projection simplifies the computation of addresses, as figure 3 shows.



An early implementation of QTM was done by Goodchild and Yang (1991) in an NCGIA hierarchical spatial data structure (HSDS) project. It used a different facet addressing scheme from the above, centering on encoding and manipulating data in the multilevel triangular raster defined by the QTM grid. Twelve- and 15-neighbor chain encoding methods were used to identify connected components and enable intersection and dilation in a triangular raster. Visualization software was also created to let users at workstations orient a globe and zoom in to regions at various levels of detail. A similar, independent project was undertaken at NASA Goddard Space Center around the same time. The sphere quadtree (SQT) of Fekete (1990) recursively decomposes facets of an icosahedron into four triangular "trixels" to spatially index geodata (satellite imagery) and manipulate it directly on the sphere. The addressing scheme for SQT is similar to that for QTM, but relations between its 20 individual trees are more complex than those of QTM's 8 facets. Algorithms for finding neighbors, connected component labelling and other basic operations were developed, but further SQT development seems not to have taken place. Besides these closely-related approaches, Lugo and Clarke (1995) have applied QTM itself to indexing and compressing digital terrain models, and Barrett (1995) -- also of NASA Goddard -- has proposed using QTM to index astronomical catalogues (looking outward from Earth rather than inward). Otoo and Zhu (1993) developed a "semi-quadcode" variant of QTM, which was claimed to be optimized for efficient spatial access and dynamic display of data on a sphere.

Many discussions of this topic tend to hang up on the choice of an initial polyhedron, and then on what subdivision method should be used. The model explored in this paper is rooted in an octahedron, for reasons that are explained next. This is not to discount any other polyhedral approaches, which also make sense given the types and constraints of the applications their models are intended to support. But the principal one with which we are concerned – managing vector GIS data for multi-resolution analysis and display – centers on identifying and negotiating conflicts among vertices and edges of digital descriptions of point, line and area features. Because the focus of this work is in the vector domain we do not maintain and manipulate (hierarchical or other) raster images of map data. As section 4 shows, we substitute QTM addresses for coordinates of map features, then analyze these codes at lower levels of resolution to simplify features and negotiate conflicts among their vertices to retrieve consistent, simplified feature descriptions at smaller scales.

1.2 Consequences of Different Polyhedral Bases

All polyhedral triangulations other than regular tetrahedra, octahedra and icosahedra have facets that are non-equilateral or vary in size and shape, becoming more diverse in size and shape as the mesh densifies. As this is a law of nature that cannot be defeated, why choose an octahedron for developing a triangular mesh on a sphere? Any of the platonic solids might do as well, or even less regular polyhedra, such as a cubeoctahedron or a rhombic dodecahedron (divided into triangles). Why does this choice matter, inasmuch as descendent QTM facets will vary in size and shape, regardless of what polyhedron one starts with? The choice may be arbitrary, but does have consequences. If we restrict ourselves just to polyhedra which have only triangular faces, each of which has the same area (but not necessarily equal angles or edge lengths), there are still quite a few to choose from, especially if truncated and stellated ones are included. The simplest case is a tetrahedron; the octahedron is the next more complex one, followed by the icosahedron, and then by various archimedean solids. Given that each facet of the original polyhedron roots a quadtree of triangles that grow smaller and more heterogeneous as they subdivide (see table 3), two consequences in particular need to be considered:

- The larger the spherical area subtended by the initial polyhedral facets, the more will the areas of sub-facets vary at any given detail level or map scale;
- The greater the number of initial polyhedral facets, the more often will map features cross facet boundaries (involve traversing two or more data trees).

These phenomena, while inevitable, are both undesirable, because each tends to complicate certain kinds of computations. The first one penalizes simpler shapes such as tetrahedra and octahedra by making the variance of error of facet areas greater than necessary. The second one complicates data management of shapes such as icosahedra by partitioning them into larger number of rooted trees, each with three edge neighbors and at least four vertex neighboring trees in the forest. These other trees must be examined when a feature or a search path crosses an edge or vertex a neighbor shares with the tree of the facet currently being traversed.

These two constraints offset one another, so that one can trade areal equality for structural simplicity. The balance to be struck could depend on the application for which QTM data is to be used. For example, an environmental sampling regime may assume equal probability of inclusion of random points in a sampling grid, which means that the areas of all facets should be as nearly equal as possible (which generally requires map projections). An early example of this approach to tessellation was the work of Wickmen et al (1974). A better-known example is documented in White et al (1992), describing the EMAP hierarchical data model developed for the US Environmental Protection Agency. Statistical considerations (minimizing distance variance between sample points) pointed toward using a basis polyhedron with a large number of facets, and the one selected (a *truncated icosahedron*, familiar as a *soccerball*, and also as the *Fullerene* carbon molecule) has 32 of them, 20 hexagons and 12 pentagons. Figure 2 shows this shape, its soccerball variant, and how it is used in the EMAP program. Orienting the polyhedron in this particular way allowed the 48

conterminous US states to fit in one hexagonal facet; but inevitably, this strategy dissected other nations and territories in inconvenient ways.

Figure 2

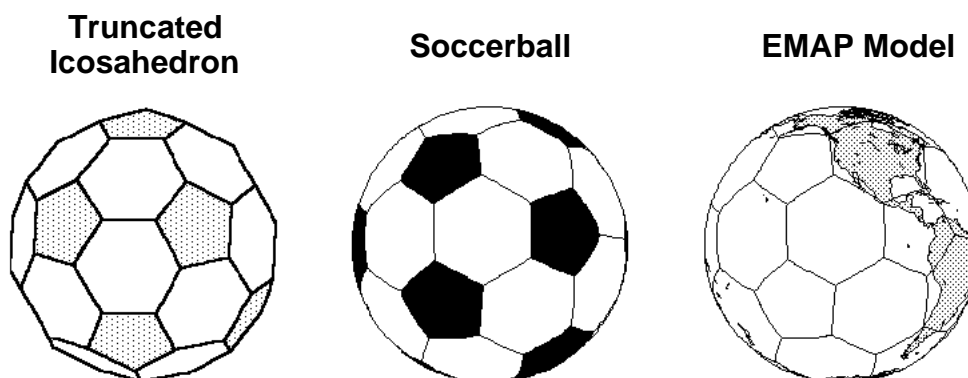


Illustration courtesy of J. Kimmerling, Oregon State University

Structural simplicity is greatest in a tetrahedron, which has 4 facets, 6 edges and 4 vertices. Each facet has 3 edge neighbors and 3 vertex neighbors (both the same), a total of 3 unique neighbors. The EMAP soccerball, on the other extreme, has a more complicated set of neighborhood relations; the hexagonal facets have 6 edge and no additional vertex neighbors, and the pentagonal facets have 5 neighbors each. The average number of neighbors is thus $(20 \cdot 6 + 12 \cdot 5) / 32$, or 5.625. An icosahedron has exactly 9 total neighbors per facet, and an octahedron has 6 neighbors.

Another way to quantify this is to compare the total number of edge neighbors, a measure of structural complexity that indexes the number of expected transitions between facets. This number ranges from 6 for a tetrahedron to 12 for an octahedron to 30 for an icosahedron to 90 for a truncated icosahedron. When one considers how relatively often it may be necessary to relate data in adjacent trees, the larger polyhedra start to look less attractive, and an octahedron may be a wise compromise, with only 12 edge adjacencies. Table 1 summarizes the parameters discussed above; *N/F* means neighbors per facet; *VN* is edge neighbors; *EN* is edge neighbors; *TN* is total neighbors.

Table 1

Basis Shape	V	E	F	VN/F	EN/F	TN/F
Tetrahedron	4	6	4	2	3	3
Octahedron	6	12	8	3	3	6
Icosahedron	12	30	20	5	3	9
"Soccerball"	60	90	32	5.625	5.625	5.625

V = Vertices; *E* = Edges; *F* = Faces

1.3 Basic Properties of the Octahedral Framework

Perhaps the most compelling reason for using an octahedron as a basis for a QTM is not its topological properties, but the fact that it can be readily aligned with the conventional geographic grid of longitude and latitude. When this is done, its vertices occupy cardinal points and its edges assume cardinal directions, following the equator, the prime meridian, and the 90th, 180th and 270th meridians, making it simple to determine which facet a point on the planet occupies. Each facet is a right spherical triangle. Except for the one at the South Pole, all vertices are located in ocean areas, minimizing node adjacency problems for most land-based geospatial data. Table 2 defines the octahedral facets when vertices are at cardinal points.

O-QTM numbers octants from 1 to 8, proceeding clockwise in the northern hemisphere from

the prime meridian, then continuing in the same direction in the southern hemisphere. One simple function that computes octant numbers is:

$$\text{OCT} = (1 + \text{LON} \text{ div } 90) - 4 * (\text{LAT} - 90) \text{ div } 90$$

This will yield an incorrect result at the South Pole but nowhere else. Negative longitudes must be complemented prior to computing octants. Given the number of an octant, it is easy to identify neighboring ones, which meet along octant edges except for the North neighbors of octants 1-4 and the South neighbors of octants 5-8, which meet only at the poles:

$$\begin{aligned} \text{EAST_NEIGHBOR}(\text{OCT}) &= 1 + (\text{OCT} + 8) \text{ mod } 4 + (4 * (\text{OCT} \text{ div } 5)) \\ \text{WEST_NEIGHBOR}(\text{OCT}) &= 1 + (\text{OCT} + 6) \text{ mod } 4 + (4 * (\text{OCT} \text{ div } 5)) \\ \text{NORTH_NEIGHBOR}(\text{OCT}) &= 1 + (\text{OCT} + 9 - 2 * (\text{OCT} \text{ div } 5)) \text{ mod } 4 \\ \text{SOUTH_NEIGHBOR}(\text{OCT}) &= 9 - (\text{OCT} + 9 - 2 * (\text{OCT} \text{ div } 5)) \text{ mod } 4 \\ &\quad - (2 * \text{OCT} \text{ mod } 2) \end{aligned}$$

The last two functions can easily be modified to eliminate vertex neighbors, if there is no need to handle such transitions. Overall, the octahedron has some useful properties and no significant disadvantages, other than generating subdivisions that have greater areal variation than do figures having a larger number of facets. An analysis and summary of QTM areal variation and its consequences is presented in section 2.

Table 2

Octant	MinLon	MaxLon	MinLat	MaxLat
1	0	< 90	> 0	90
2	90	<180	> 0	90
3	180	<270	> 0	90
4	270	<360	> 0	90
5	0	< 90	-90	0
6	90	<180	-90	0
7	180	<270	-90	0
8	270	<360	-90	0

2. Areal Inequalities in the O-QTM Tessellation

We embed QTM's initial octahedron in a sphere by placing its six vertices on the surface of a unit sphere (which may readily be scaled to Earth radius), such that the distance of the six octa vertices from the center of the octahedron (sphere) is unity. All other points (along octa edges and within octa faces) lie closer to the center than do the vertices, with the centroids of the eight facets lying at the smallest radius enclosed by the octahedron. As the QTM structure develops, it blossoms into a multifaceted polyhedron, having 32, 128, 512, 2048 ... faces. This section discusses the size and shapes of these planar facets, all vertices of which touch a circumscribed sphere, but whose edges are geodesic lines (chords through the sphere).

Unlike geodesic domes and related polyhedral world models such as described in section 1, O-QTM does not follow great circles on the sphere in decomposing facets. When a facet is subdivided, the latitudes and longitudes of pairs of its vertices are averaged to yield edge midpoint locations. Except along edges of the original octahedron (lat or $\text{lon} \text{ mod } 90 = 0$), these midpoints do not coincide with locations on great circles that connect existing vertices. Bisecting a facet edge with a normal vector, then extending the normal to the surface will also bisect the latitudes and longitudes defining the endpoints of that edge *only* along great circles, not along small ones (parallels). In addition, halving latitudes and longitudes creates a tessellation with certain asymmetries; east-west edges -- being parallels -- are straight, but the other two sets of edges are not straight and only roughly parallel. As a result, most of the triangles in the QTM network have different shapes and larger areas than would spherical triangles defined from the same set of vertices. Still, the chord length of any given

edge segment will be identical to that of a great circle passing through the same endpoints, and can be computed using a standard formula (Snyder, 1987):

$$\sin(\text{arc}/2) = \{\sin^2[(\text{lat1}-\text{lat2})/2] + \cos(\text{lat2})\cos(\text{lat1})\sin^2[(\text{lon2}-\text{lon1})/2]\}^{1/2}$$

The chord length (in radians) is twice the value of this expression. Using this result, the areas of facets can be tabulated via the formula:

$$\text{Area} = (s(s-a)(s-b)(s-c))^{1/2}, \text{ where } a, b, \text{ and } c \text{ are the chord lengths, and } s = (a+b+c)/2 \text{ (half the perimeter).}$$

Plane facet size statistics based on these formulae for the first five QTM levels are presented in Table 3. Note how the total area of the figure rapidly approaches that of a sphere, and that the ratio of maximum to minimum facet area stabilizes at 1.83... Other indicators of uniformity are the decline of normalized deviations with level, and the decline of kurtosis values at higher levels from that of a normal distribution to values characterizing a uniform one (Bulmer, 1967, p. 64).

Table 3

Area property	Level 1	Level 2	Level 3	Level 4	Level 5
Facets *	32	128	512	2048	8192
Min Area	0.28974	0.07554	0.01917	0.00481	0.00120
Max Area	0.43301	0.13188	0.03466	0.00877	0.00220
Max/Min Area	1.49448	1.74583	1.80803	1.82328	1.83333
Mean Area	0.32555	0.09346	0.02424	0.00612	0.00153
Median Area	0.28974	0.09342	0.02387	0.00611	0.00152
Std Dev	0.07164	0.01612	0.00354	0.00083	0.00020
Kurtosis	1.74493	2.91530	2.99556	2.44369	2.15422
Total Area	10.41775	11.96281	12.41066	12.52712	12.55654
Sphere Area	12.56637	12.56637	12.56637	12.56637	12.56637
% of Sphere	82.90183	95.19705	98.76091	99.68765	99.92174

* Statistics for columns are based on $n = \text{Facets}/8$ (single octant).

3 Computational Properties of QTM Identifiers

An O-QTM location code consists of an octant number (from 1 to 8) followed by up to 30 quaternary digits (from 0 to 3) which name a leaf node in a triangular quadtree rooted in the given octant. Central facets have an ID of 0, and corner facets take the ID of the vertex defining them; when a vertex appears, its number is assigned as six minus the sum of the IDs of the endpoints of the bisected edge, as figure 3 shows. Each QTM digit doubles linear precision, identifying a specific facet having slightly more than one-fourth the area of its parent. For example, the 18-level QTM ID for the building housing the Geography Department at the University of Zürich is 1133013130312301002; this encodes the geographic location 47° 23' 48" N, 8° 33' 4" E within about 60 meters (roughly the length of the building, and close to the limit of precision obtained from measuring on a 1:25000 scale topographic map), occupying a QTM facet about 1,000 meters square. Finer resolution, when available, is expressed by adding low-order quaternary digits to the ID; 30 digits will encode locations to an accuracy of about 2 cm., and can be expressed as one 64-bit word (Dutton, 1996). Note that these identifiers have a different sequence of quaternary digits in Goodchild and Yang's O-QTM system, and are different still in Fekete's I-QTM system (the former would identify the same triangular facet, giving it a different name; the latter would identify a facet having entirely different vertices and which would be 60% smaller in area for a given number of digits).

One basic operation of any geocoding scheme is determining proximity of points. To date no complete "native" solution for determining distance between two QTM IDs has emerged, but heuristics have been developed for closely-related computations. For example, one often

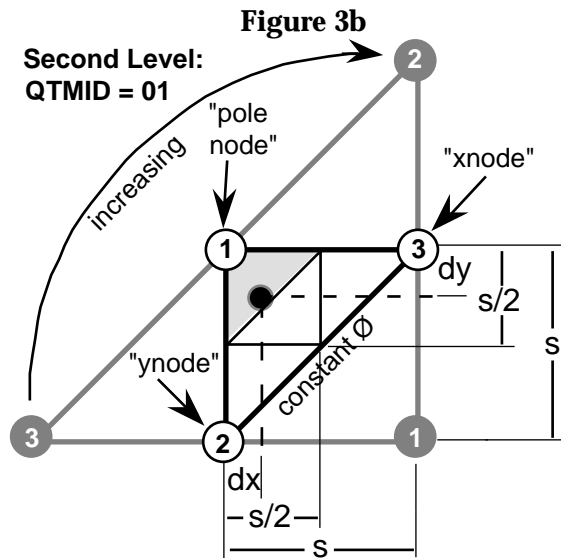
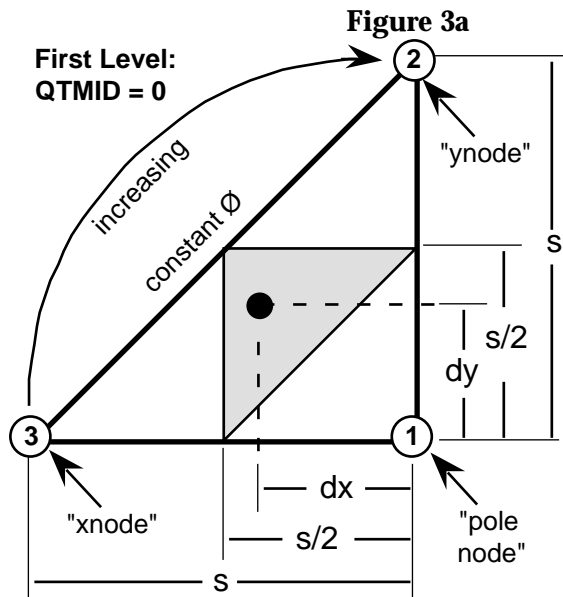
needs to know if two QTM facets share an edge or vertex at some level of detail. There are three distinct cases:

1. Both facets are interior cells (both IDs terminate in 0)
2. One facet is an interior cell (its ID terminates in 0)
3. Neither facet is an interior cell (neither ID terminates in 0)

The first case rules out edge adjacency, as 0 cells only have non-zero neighbors (but not conversely). The second case is also trivial, because any 0 cell's neighbors will also be its siblings, and their IDs will be identical except for the last digit. The third case is more interesting; while we cannot easily affirm that two such facets are neighbors, it is easy to identify many cases where they are not. First, the QTM numbering scheme guarantees that all facets that share a common vertex must have the same terminal digit, either 1, 2 or 3 (six facets will share a vertex except at the six vertices of the initial octahedron, where four do). Second, two facets will be adjacent only if their QTM IDs differ in exactly one digit, and as the prior sentence asserts, this cannot be the terminal digit as long as case 2 is not true (this property is also one exhibited by Fekete's SQT model). Unfortunately, the converse is not true; many IDs that differ in only one digit are not adjacent. Regardless, if more than one digit of two IDs differs, one can be sure the two facets are not neighbors. A geometric (rather than lexical) approach to solving this problem is described in section 3.2.

3.1 Conversion from Geographic Coordinates to O-QTM IDs

Computing a quaternary address for a geographic location involves recursively identifying child facets of the octant containing the location, and giving each one visited an appropriate name from the set {0,1,2,3}. Goodchild and Yang (1992) project octants to equilateral triangles, and apply algorithms that require exponentiation and irrational arithmetic. The approach used here projects to right isosceles triangles, and requires only linear arithmetic, which could be performed on properly-scaled integers. The Zenithial OrthoTriangular (ZOT) map projection (Dutton, 1991) is used to project the 8 octants to a square; the North pole occupies the center, and the South pole maps to the four corners (see fig. 1b). Each octant is a right isosceles triangle, as is each QTM quadrant at every level of detail. ZOT space uses *Manhattan Metric*, so that distances are always the sum of x- and y-displacements. Figure 3 illustrates this method across two levels of detail.



To encode a QTM ID in Manhattan space, test whether the point's X and Y offsets sum to less than half the side length; if so, return number of Pole Node. If either X or Y is greater than half the side, return the number of that node; else return 0.

s = Facet side length in ZOT metric
 $s/2$ = half side length
 $dy = \varnothing$; latitude change from origin
 $dx = -dy$; other coordinate
 If $(dx+dy) < s/2$ then return (1);
 If $dy > s/2$ then return (2);
 if $dx > s/2$ then return (3);
 else return (0);

New nodes in the triangular mesh appear at edge midpoints and are assigned numbers by subtracting the sum of the nodes of the bisected edge from 6, as figure 3b shows.

3.2 Conversion from O-QTM IDs to Geographic Coordinates

It is possible to directly convert a QTM ID back to latitude and longitude at any level of detail it contains, within the accuracy that level of detail denotes. The location computed is an arbitrary point within a triangle, normally chosen as its centroid. One method for doing this uses the same algorithm described in figure 3, computing triangle vertices in ZOT space, but skipping the point-in-triangle test, as a QTM ID specifies what triangle to describe next. The ZOT centroid of the leaf facet is then de-projected into latitude and longitude. A more direct and faster method to compute centroids of facets at the desired level of detail, uses a 3-axis coordinate system. The axes connect octant vertices to their opposite edge midpoints (with a local origin at 30°N/S, 45°E/W) to define locations in each of the spherical right triangles, as figure 4 shows. Starting at the root, each digit is mapped to movement along one of these axes. However, because the central cell of each group of four siblings is always coded as 0, encountering a 0 digit denotes no movement, only a reversal of future direction along all axes. Movement along axes are accumulated in three registers, then projected to the North-South ("1") axis and scaled to yield a latitude, which is used in computing longitude. The following function definitions summarize this approach:

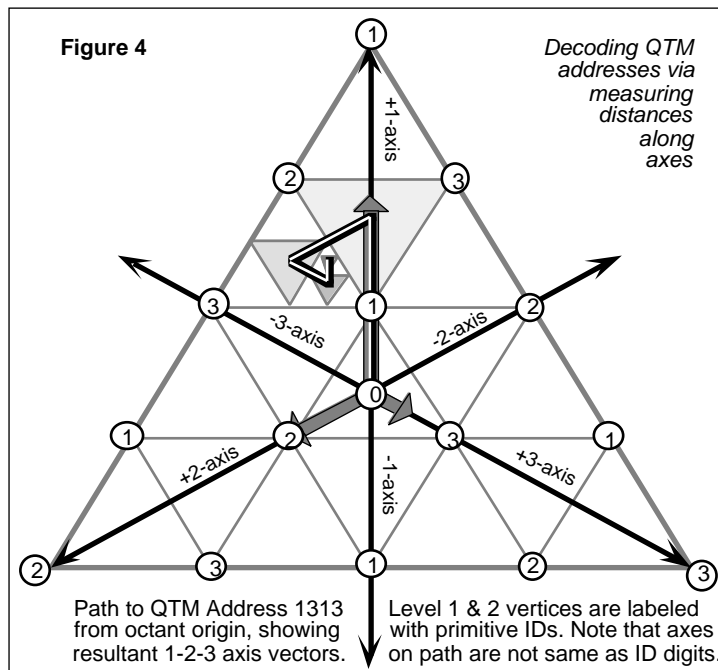
```
VOID QTMtoAxes(QTMID qid, INT qlen, INT level, BOOL vertex, REAL axes[3])
```

The function returns no value, placing its result in *axes*; it will compute a meaningful set of axial distances from any valid set of input parameters. The role of the *vertex* parameter is discussed below. The *axes* are converted to a latitude and a longitude by the functions:

```
REAL AxesToLat(REAL axes[3])  

REAL AxesToLon(REAL axes[3], lat)
```


In *QTMtoAxes*, a distance parameter is initialized at 1/12 the mean circumference of the Earth, and is divided in half at each level. The same parameter is used in moving along each of the three axes. However, were we to model an ellipsoid rather than a sphere, we could generalize this method by implementing a *vector* of three distance steps (one for each axis, scaled according to an ellipsoid model). The *axes vector* that *QTMtoAxes* returns is then used to compute first a latitude, then a longitude for the QTM ID being evaluated, at the specified level of detail.



We now return to the problem of determining if two QTM IDs represent neighboring facets or not. We wish to evaluate vertex as well as edge neighbors, because applications we pursue (for example, cartographic generalization) must identify nearby points that may not lie in the same quadrant or whose QTM IDs do not share a parent (even a common ancestor). The above functions can do this when the *vertex* parameter of *QTMtoAxes()* is set to TRUE. Then, (unless the ID ends in zero) one more statement is executed. This causes the location of the vertex named by the last ID digit to be returned rather than the centroid of the facet (zero-coded (central) facets are not associated with any vertex

other than their centroids, and thus have no vertex neighbors). While the axis vectors may differ considerably, they will evaluate to the same latitude and longitude for each of the six facets that share this vertex. Effects of round-off errors can be cancelled by comparing returned pairs of latitudes and longitudes to one another within a tolerance related to QTM level of detail. Locations in abutting octants require no special handling if longitudes are offset and axes initialized appropriately in each octant.

4 Application to Cartographic Generalization

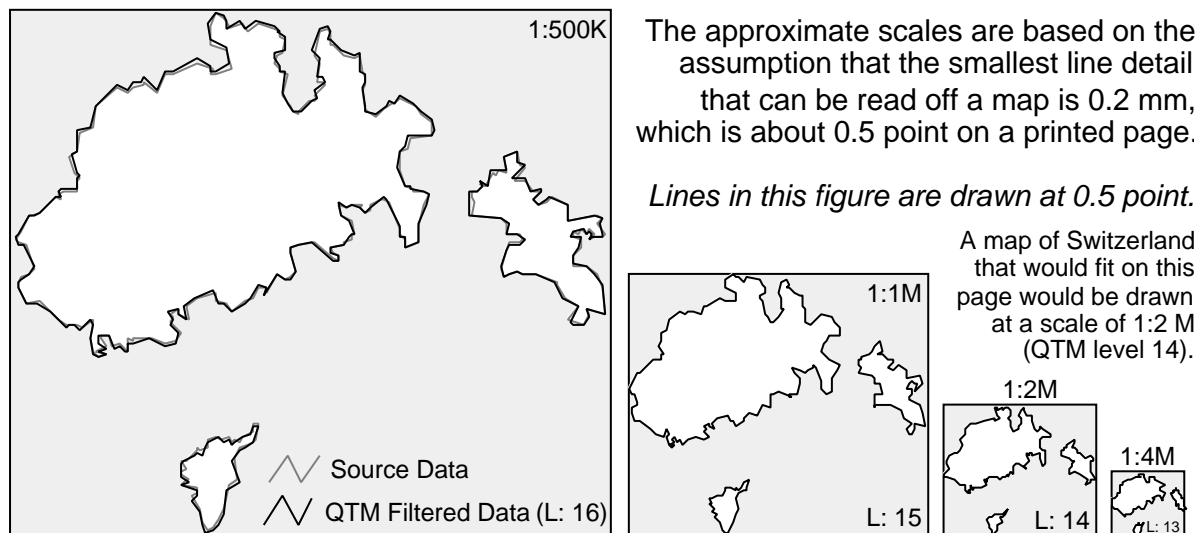
The information content of QTM IDs is greater than the coordinates they represent, because the model denotes scale and accuracy as well as position. By encoding each coordinate of a cartographic database into QTM at levels of resolution appropriate to the source data (and these can change from one layer, one feature or even one vertex to another), a *multi-resolution representation* results. We consider this to be a potentially more efficient alternative to storing *multiple representations* of a set of features (i.e., a separate database for each scale), because QTM provides a unified description of model data that is potentially capable of rendering that data across a range of scales. It is also easier to maintain and edit cartographic features stored in a unified description than a set of multiple representations would be.

Current research involves exploring the above and other algorithms in a feature-based cartographic data processor that receives source data from GISs via cartographic exchange file formats, encodes features or layers into QTM representations, and extracts this data at source and smaller scales to test how well this concept of space handles map generalization. As described in Dutton (1996) and Dutton and Battenfield (1993), QTM encoding provides a number of properties beyond hierarchical geocoding that can help identify and potentially assist in resolving conflicts for map space because of scale change. This requires additional data structures to achieve, but perhaps not as many as current approaches, such as Jones et al (1992), which uses a non-hierarchical triangulation across feature classes. This and other object-based methods model explicit relationships of map geometry (e.g., by topological

simplices). Our approach models space rather than objects; it exploits implicit relationships that location identifiers denote within a multi-resolution global spatial reference framework.

Figure 5 illustrates an early result of using QTM to generalize vector map features. This set of polygons (the Swiss canton of Schaffhausen) had been previously filtered via the Douglas algorithm to 306 segments, of mean length 0.56 km. Analysis of this data indicated that all its details could be encoded in 18 QTM digits, so that it is represented at roughly 1:100 000.

Figure 5: QTM Generalization Results at Four Scales
Schaffhausen Canton, Switzerland



The QTM-filtered representations decrease detail to 300 points (0.58 km) at QTM level 16, to 290 points (0.59 km) at level 15, to 258 points (0.65 km) at level 14, and to 200 points (0.79 km) at level 13. Typically, the filtering process results in little point reduction near the limit of resolution, with more changes appearing 2 or 3 levels down, then tapering off again after 5 or 6 detail levels. The useful range of detail depends on various aspects of both the digitized data and its application, but would not normally exceed a ratio of about 1:250. Also, more sophisticated approaches to retaining, displacing and eliminating polyline vertices in the QTM domain are possible than the experiment illustrated above used, and will be explored during the course of our project.

References

- Barrett, P. (1995). *Application of the linear quadtree to astronomical databases*. **Astronomical Data Analysis Software and Systems IV**, ASP Conf. Series, v. 77.
- Bulmer, M.G. (1967). **Principles of statistics**, 2nd ed. Cambridge MA: MIT Press.
- Dutton, G. *Geodesic modelling of planetary relief*. **Cartographica**. Monograph 32-33, Vol. 21 no. 2&3, 1984; pp. 188-207.
- Dutton, G. *Modelling locational uncertainty via hierarchical tessellation*. **Accuracy of Spatial Databases** (M. Goodchild & S. Gupta, eds). Taylor & Francis, 1989. pp. 125-140.
- Dutton, G. *Zenithial Orthotriangular Projection*. **Proc. Auto-Carto 10**. Falls Church, VA: ACSM/ASPRS, 1991, pp. 77-95.
- Dutton, G. & B.P. Buttenfield. (1993). *Scale change via hierarchical coarsening: Cartographic properties of Quaternary Triangular Meshes*. **Proc. 16th Int. Cartographic Conference**. Cologne, Germany, pp. 847-862.
- Dutton, G. (1996) *Improving locational specificity of map data - a multi-resolution, metadata-driven approach and notation* **Int. J. GIS**, 10:3, pp. 253-268.
- Fekete, G. (1990). *Rendering and managing spherical data with sphere quadtree*. **Proc.**

Visualization '90. New York: ACM.

Fisher, I. and O.M. Miller (1944). **World Maps and Globes.** New York: Essential Press.

Fuller, R.B. (1982). **Synergetics: Explorations in the geometry of thinking.** New York: Macmillan, 2 vols.

Goodchild, M. and Yang Shiren (1992). A hierarchical data structure for global geographic information systems. **CVGIP**, 54:1, pp. 31-44.

Gray, R.W. (1994). Fuller's Dymaxion Map. **CaGIS** 21:4, pp. 243-246.

Jones, C.B., J.M. Ware and G.L. Bundy (1992). Multiscale spatial modelling with triangulated surfaces. **Proc. Spatial Data Handling** 5, vol. 2, pp. 612-621.

Lugo, J.A. and Clarke, K.C. (1995). Implementation of triangulated quadtree sequencing for a global relief data structure. **Proc. Auto Carto 12.** ACSM/ASPRS, pp. 147-156.

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

Snyder, J.P. (1992). An equal-area map projection for polyhedral globes. **Cartographica** 29:1, pp. 10-21.

Snyder, J.P. (1987). *Map Projections - A Working Manual.* **US Geological Survey Prof. Paper 1395, Washington: USGPO.**

Unknown (1943). R. Buckminster Fuller's Dymaxion World. **Life**, March 1.

White, D., J. Kimmerling, and W.S. Overton (1992). Cartographic and geometric components of a global sampling design for environmental monitoring. **CaGIS**, 19:1, pp. 5-22.

Wickman, F.P., Elvers, E. and Edvarson, K. (1974). A system of domains for global sampling problems. **Geografisker Annaler**, 56A 3-4, pp. 201-211.