Zenithial Orthotriangular Projection

A useful if unesthetic polyhedral map projection to a peculiar plane

Geoffrey Dutton[1]
Spatial Effects
150 Irving Street
Watertown, MA 02172 USA
qtm@cup.portal.com

Abstract

This paper describes the construction, properties and potential applications of a cartographic projection recently developed by the author, called the *Zenithial Orthotriangular (ZOT) projection of an Octahedron. ZOT* maps a planet to a plane by modelling it as an octahedron (a regular solid having 8 equilateral triangular facets), which is then unfolded and stretched to fit within a square. As described below, *ZOT* is developed from a regular octahedron mapped in North polar aspect, by cutting octant edges of the southern hemisphere from pole to equator, and stretching all octahedral facets to occupy eight identical right triangles (extensions to the ellipsoid are described). The North pole lies at the center of projection, while the South Pole occupies all four corners; points along map borders are mirrored across the central axes. After discussing its cartographic properties, *ZOT*'s relation to the Quaternary Triangular Mesh (*QTM*) global tessellation is explored. The use of *ZOT* is shown to facilitate recursive definition of *QTM*'s geodesic graticule of nested triangles. Computationally, this structure is handled as a quadtree, even though its elements are triangular in shape. Basic procedures for mapping geographic coordinates to *QTM* quadtree addresses via *ZOT* are presented and discussed, and suggestions given for standardizing how *QTM* tiles are addressed in *ZOT* space.

[1] The author gratefully acknowledges encouragement and support for this work from Prime Computer, Inc.

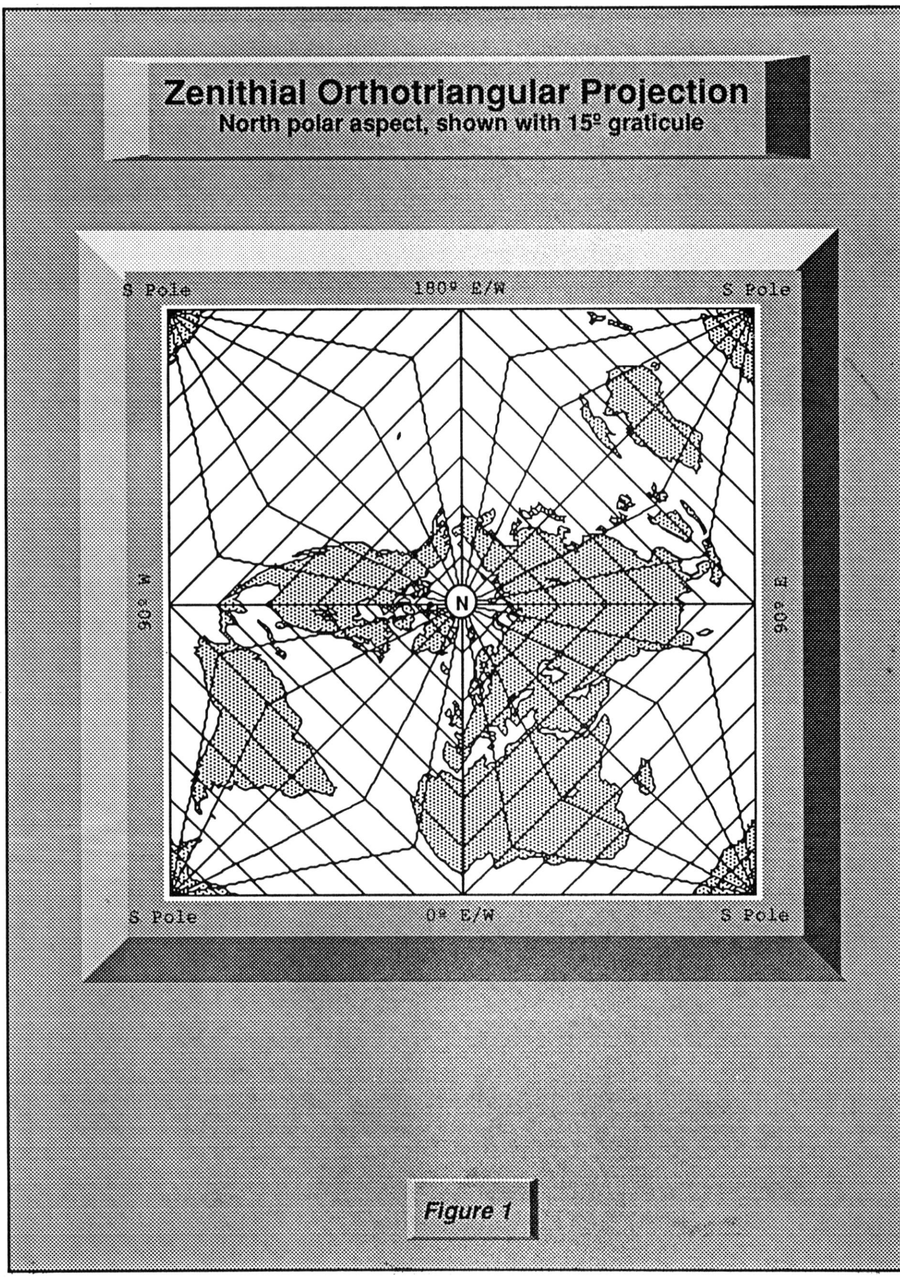
Polyhedral Maps

There is a family of maps called *polyhedral projections* that apportion regions of Earth to coincident facets of some concentric polyhedron. If the polyhedron is one of the five platonic solids, these facets will be either square, pentagonal or most likely, triangular, and all the same size and shape. While these figures may be torn apart and unfolded in a number of ways, no regular polyhedron beyond the tetrahedron can be unfolded to lie on the plane in a maximally compact way; there will always be concavities whatever arrangement of facets is used. As a consequence, polyhedral maps tend to have convoluted, lobed shapes, rather than fitting neatly into a rectangle, as do most projections. This apparently frustrates cartographers, who often seem to feel that polyhedral projections involve excessively complicated computational procedures. This is only partly true: however odd and enigmatic such constructions may be, they are at least *regular* and *enumerable.*

Mapping regions of the Earth to facets of a polyhedron can involve any of a number of map projections, the most natural of which is the *gnomic.* This is one of the few projections in which all coordinates relate to a single point of reference (the center of the planet). Although gnomic projections are not suitable for large areas, their distortions are quite minor when limited to the facets of enclosing polyhedra. Most azimuthal projections (such as the stereographic) require multiple reference points in order to portray the entire globe. This paper describes an azimuthal mapping of of an *octahedron* to a *square* in North polar aspect.

Projective Properties

The *ZOT* projection is *zenithial* (azimuthal) because meridians remain straight and of constant radial spacing; longitudes may be measured directly with a protractor. There is, however, more than one azimuthal origin, as longitudes are only true within a hemisphere. As the South pole is separated into four locations, meridians in the southern hemisphere originate at each of the four corners of the projection. *ZOT* also has the *equidistant* property; distance between parallels is constant throughout the map. The projection has been named *orthotriangular* because it maps spherical triangles to right triangles in its domain. These properties are evident in the world map in ***Figure 1***. *ZOT* is also *doubly periodic;* that is, it may be repeatedly tiled in two directions to fill a plane, as ***Figure 2*** illustrates.


Zenithial Orthotriangular Projection
North polar aspect, shown with 15º graticule
S Pole
180º E/W
S Pole
90º W
N
90º E
S Pole
0º E/W
S Pole
Figure 1

Zenithial Orthotriangular Wallpaper

North and South polar aspect, shown with 15º graticule

ZOT is doubly periodic (repeating in X and Y)

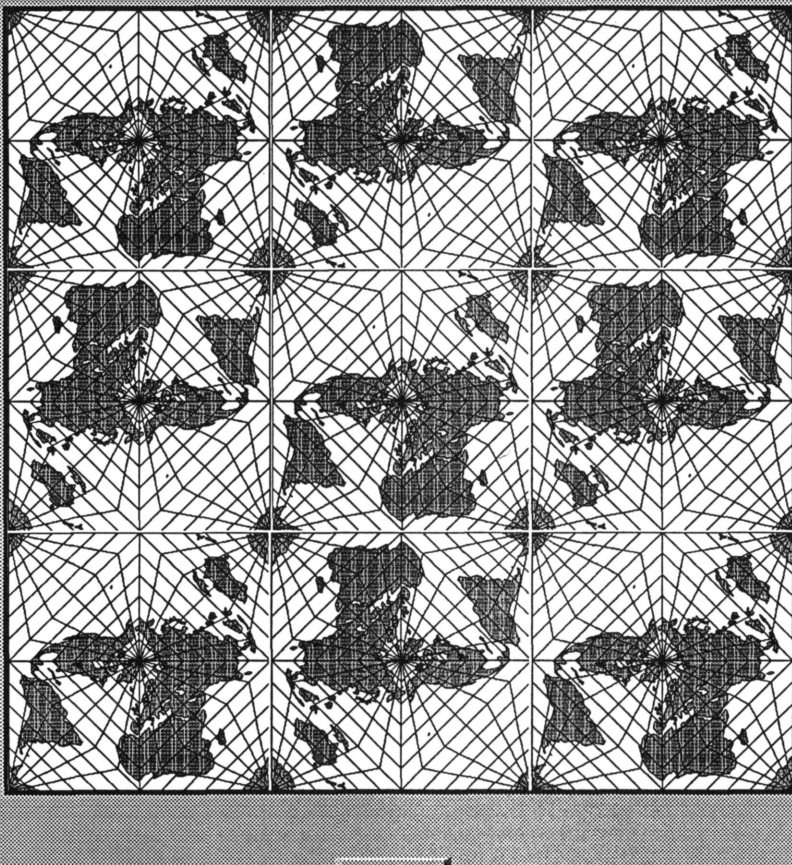


Figure 2

ZOT is neither *equal-area* nor *conformal*. Along parallels, map scale varies inversely with latitude, with the error factor growing from unity at the pole to $\sqrt{3}$ at the equator. This occurs because the equilateral facets of the octahedron are mapped to right triangles, causing their equatorial bases to expand. Along any given meridian, map scale is constant. However, the scale varies linearly from one meridian to the next, from unity (at 45, 135, -135 and -45 degrees) to $\sqrt{2}$ (at 0, 90, 180 and -90 degrees longitude), cycling four times around the equator. In general, there is no scale error at the poles, a small amount in the vicinity of the 8 octa face centers and more near their edges, being greatest along the four equatorial edges, and increasing toward the four equatorial vertices (which occupy the midpoints of *ZOT* map margins).

Despite this variability, all meridians map to straight lines which flex at the equator, and parallels to straight lines which flex at each 90th meridian, due to the piecewise continuous (polyhedral) nature of the projection. In most polar azimuthal projections, parallels map to circles or ellipses. In the orthotriangular projection, they map to diamonds (squares). This derives from the distance metric ("Manhattan") employed, and reflects the fact that the projection maps a sphere to the planar facets of an octahedron. This rectalinearity and modularity makes the projection very easy to compute, as it permits geographic coordinates to be mapped to the plane using linear equations, without recourse to trigonometric formulæ, square roots or, under restricted conditions, real arithmetic.

One obvious, even disturbing, property of ZOT is the 90º change in direction of parallels at every 90th meridian. This causes strange distortions in the shapes in all major land masses other than South America and Australia. Likewise, the flexing of meridians at the equator distorts Africa and South America. The former effect can be minimized by offsetting meridional octant edges roughly 25º to the West, which bisects land masses at more natural locations. The latter effect cannot be mitigated, as the equator cannot be shifted in any useful way. For computational purposes ZOT's orientation is rather immaterial, but should be standardized (see suggestion below).

Computing ZOT Coordinates

When a point is to be projected, its colatitude is multiplied by the map scale; the product is multiplied by the point's longitudinal displacement from the left edge of the octant and divided by $\pi/2$. The result is either an x or y offset from the pole's location, depending on the octant within which the point lies. We compute the other offset by subtracting the first one from the scaled colatitude; this fully determines the point's x,y location on the map.*The procedure's simplicity derives from using "city block" distances (Manhattan Metric), in which distance between points is the sum of x and y displacements, instead of Pythagorean distances.* In other words, all points along a given *ZOT* latitude are equidistant from the pole closest to them (the sum of x and y is constant and proportional to colatitude). The locus of all points along a given latitude is a straight line cutting through the octant at 45º (parallel to its equatorial base); a given distance traversed along a parallel has a size proportional to longitude, another simple linear function. The *ZOT* projection for the North polar aspect may be derived as follows:

Derivation of ZOT x,y coordinates from geographic Locations

Given:

```
double Plat     := Latitude being projected  -- In Radians
double Plon     := Longitude being projected -- In Radians
double Diam     := Map diameter          -- Cm, inches or other linear unit
double S        := Diam / π              -- Absolute scale factor
double P2       := π / 2                 -- Constant for right angle
```

Parameters:[2]

```
int    OCT      := Octant occupied by point -- 1–4 in N, 5–8 in S Hemi
double R[1]     := X-coordinate Scale factor -- Sign only varies by octant
double R[2]     := Y-coordinate Scale factor -- Sign only varies by octant
double C[1]     := X-coord origin for octant -- Center, left or right side
double C[2]     := Y-coord origin for octant -- May be center, top or bottom
int    FLOPS[8] := {1, 1, -1, -1, -1, 1, 1, -1}  -- Meridional edge orientations
```

Set up Octant:

```
int    ORG      = (P2 - Plat) div P2         -- Map origin (0 = center, 1 = corner)
int    OCT      = (ORG + 1) * (Plon div P2)   -- Octant occupied (1–8)
int    X1       = 2 - ((OCT + ORG - 1) mod 2) -- 1 if Lat maps to X, 2 if to Y
int    X2       = 3 - X1                     -- 2 if Latitude maps to X, 1 if to Y
int    HS       = 1 - (2 * ORG)              -- Hemisphere Sign (1 in N, -1 in S)
double R[X1]    = S * FLOPS[OCT]             -- X or Y factor (-R left, +R right)
double R[X2]    = - S * HS * FLOPS[9 - OCT]  -- Y or X factor (-R top, +R bot)
double C[X1]    = - ORG * R[X1]              -- X or Y Center (Zero in N hemi)
double C[X2]    = - ORG * R[X2]              -- Y or X Center (Zero in N hemi)
```

[2] The parameters and variables in this algorithm are typed according to their basic cardinalities. Certain *int* parameters are also used in floating point expressions (performed in *double* precision, we presume); ints to can be converted to real as one's programming environment may require.

Project Point:

```
double  CLP    = P2 - (HS * Plat)          -- Absolute Colatitude of point
double  OLP    = CLP * (Plon mod P2) / P2  -- Long offset (prop. to Colat)
double  PX     = R[X1] * abs(CLP - OLP)    -- Relative X or Y offset
double  PY     = R[X2] * OLP               -- Relative Y or X offset
PX             = C[X1] + PX                -- Projected X Coordinate
PY             = C[X2] + PY                -- Projected Y Coordinate
```

After initial octant setup calculations (which involve computing only 9 numbers and, in most cases, need be done but a few times for a given set of coordinates), the above algorithm uses 4 additions, 4 multiplications, 1 division and 2 rational function calls to map one point from the sphere to the plane. In situations where the octant points occupy changes frequently, setup can be table-driven based on an octant number, just as table FLOPS provides signs of scale factors and axis origins.

Note that while the above algorithm assumes a spherical Earth, its principle can also be applied to ellipsoids, at the expense of some additional arithmetic. Table FLOPS represents lengths (unity) and orientations (sign) of edges of an octahedron enclosing the planet. Were this object to have non-uniform semiaxes, the entries in FLOPS would have values differing slightly from unity; this data could be used to anchor the projection to any specified ellipsoid. In the spherical case, one computes Y coordinates along a line having its intercept at *Plat* and a slope of unity, scaling X from *Plon*; for ellipsoids, the procedure involves slopes differing slightly from unity, but is otherwise handled identically to those more complex cases.

Related Antecedents

The *ZOT* is not the first world projection into a square domain having double periodicity, nor is it the first to exploit the geometry of the octahedron. It apparently is the first to employ a *Manhattan* distance metric, and one of the few which can be constructed without trigonometric functions (such as the *Peters* or *equirectangular*). One of its more interesting predecessors is the *Quincuncial* projection, developed in the 1870's by Charles Sanders Peirce. Based on elliptic integrals, this remarkable and elegant construction is *conformal* and *doubly periodic*.[3] Despite its obvious octahedral symmetry, Peirce apparently never related his projection to polyhedra. Although widely appreciated, it fell into disuse, although the Coast and Geodetic Survey used it in a 1947 world navigation map (Eisele, 1963).

[3] *Quincunx* is a Latin word meaning "arrangement of five things." Peirce's *Quincuncial* projection is just that, as it places the South pole at the corners of a square and the North pole at its center.

Also related to *ZOT* is Cahill's *Butterfly* projection (Fisher and Miller, 1944), an interrupted conformal projection of the globe onto eight triangular facets arranged in a butterfly-like shape. In each of its octants, the equator and central meridian are straight and all other meridians and parallels bow outward. As a result, assembly of the *Butterfly* results in a lumpy shape somewhere in between an octahedron and or a sphere. Also, indexing map locations is complicated both by the mathematics required for the *Butterfly* projection and the arrangement of its facets.

Buckminster Fuller's *Dymaxion* projection dates from the 1940's and seems to have undergone a metamorphosis from an initial cuboctahedron basis[4] to the icosahedral form of the version currently marketed (Life, 1943; Fisher and Miller, 1944; Fuller, 1982). Fuller's and Cahill's motivations seem to have been similar in producing these projections; to minimize scale errors and to exploit polyhedral geometry to produce a globe that can be folded from a single sheet of paper. Fuller was keen on using his projection to convey thematic data about "Spaceship Earth", (he envisioned a large Dymaxion geodesic globe studded with computer-controlled miniature lamps to depict global statistical data, but seems never to have done this). Most versions of the *Dymaxion* employ gnomic projections.

The *"polygnomic"* world projection onto an icosahedron may have first been realized by Fisher (Fisher, 1943), even though Fuller enjoyed taking credit for it. Indeed, the idea (if not its execution) can be traced back to the work of Albrecht Dürer in the sixteenth century (Fisher and Miller, 1943, p. 92). This invention suited Fuller's purposes perfectly, as it represents chords of great circles with straight lines, like the struts of one of his geodesic domes. *ZOT,* however, is *not polygnomic;* it is oriented to the poles, not to the center of the Earth. Consequently, most great circles are not straight lines in *ZOT* space (but the equator and all meridians are).

[4] A cuboctahedron is a 14-sided polyhedron having 8 triangular and 6 square facets. Unlike the five regular polyhedra, the facets are tangent to two concentric spheres, complicating construction or calculation of features that cross facet edges.

Error Adjustment

Nearly any area or distance measured from an *ZOT* projection will be incorrect by as much as a factor of two. As it is almost as simple to calculate the scale error at any point as it is to compute coordinates, and only slightly harder to derive the error involved when distances between points or polygonal areas are computed (with cases involving more than one octant presenting the most complexity). This means that size and distance calculations may be corrected as required; the greater the precision, the greater the cost. Tables can be developed to facilitate such corrections.

Polyhedral Addressing

ZOT is not esthetically pleasing, especially in comparison to the sweeping curves of Peirce's *Quincuncial*. *ZOT* generates angular discontinuities at octant boundaries, violating a number of cartographic precepts. No claim is made for it as an optimal visual matrix for presenting global spatial data. Still, *ZOT projection may have considerable computational utility* when applied to tessellated polyhedra embedded in a well-defined spherical manifold, as the following section explains.

The best uses for *ZOT* may be those which capitalize on its computational simplicity. In particular, there is a strong affinity between *ZOT* and the geometry of the Quaternary Triangular Mesh (*QTM*) global location coding model (Dutton, 1989; Goodchild and Yang, 1989). ***Figure 3*** and ***Figure 4*** illustrate how *QTM*'s recursive subdivision of octahedral facets into four tiles each is mapped to a completely regular mesh of right triangles when projected via *ZOT*. This mesh densifies in the same manner as a rectangular quadtree does, but also includes diagonal elements (parallels of latitude). Note how each triangle's edges split in half, and how its hypotenuse follows a particular latitude. This may be exploited to derive *QTM* facet addresses from latitude and longitude, as ***Figure 5*** shows.

The arithmetic used in this procedure consists of testing sums and differences of x and y displacements against one parameter (*s/2* in fig. 5) that is constant for all *QTM* tiles at a given level of detail. In addition, the algorithm needs to know the "basis number" of each node (vertex) in the *QTM* network in order to assign a *QTM* ID to every tile in the hierarchy; each vertex is identified with a 1-node, 2-node or 3-node (its basis number), and all higher-level nodes at a particular location continue to manifest its original basis number. This digit is common to all four *QTM* cells surrounding each octa vertex, and all six cells that surround the nodes that appear in subsequent subdivisions. Central (0) cells are associated

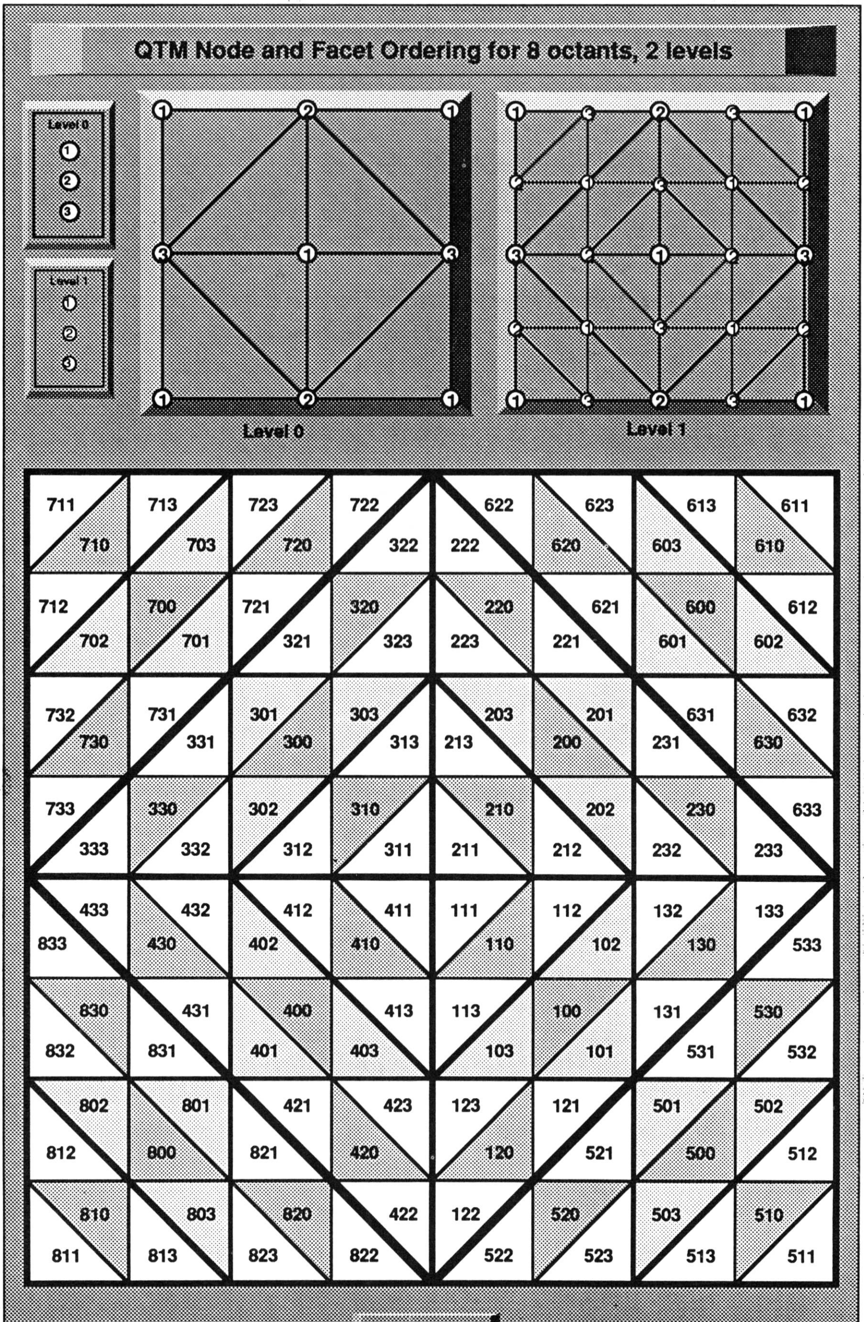

QTM Node and Facet Ordering for 8 octants, 2 levels
Level 0
Level 1
711 710 713 703 723 720 722 322 622 222 623 620 613 603 611 610
712 702 700 701 721 321 320 323 220 223 621 221 600 601 612 602
732 730 731 331 301 300 303 313 203 213 201 200 631 231 632 630
733 333 330 332 302 312 310 311 210 211 202 212 230 232 633 233
433 833 432 430 412 402 411 410 111 110 112 102 132 130 133 533
830 832 431 831 400 401 413 403 113 103 100 101 131 531 530 532
802 812 801 800 421 821 423 420 123 120 121 521 501 500 502 512
810 811 803 813 820 823 422 822 122 522 520 523 503 513 510 511

Figure 3

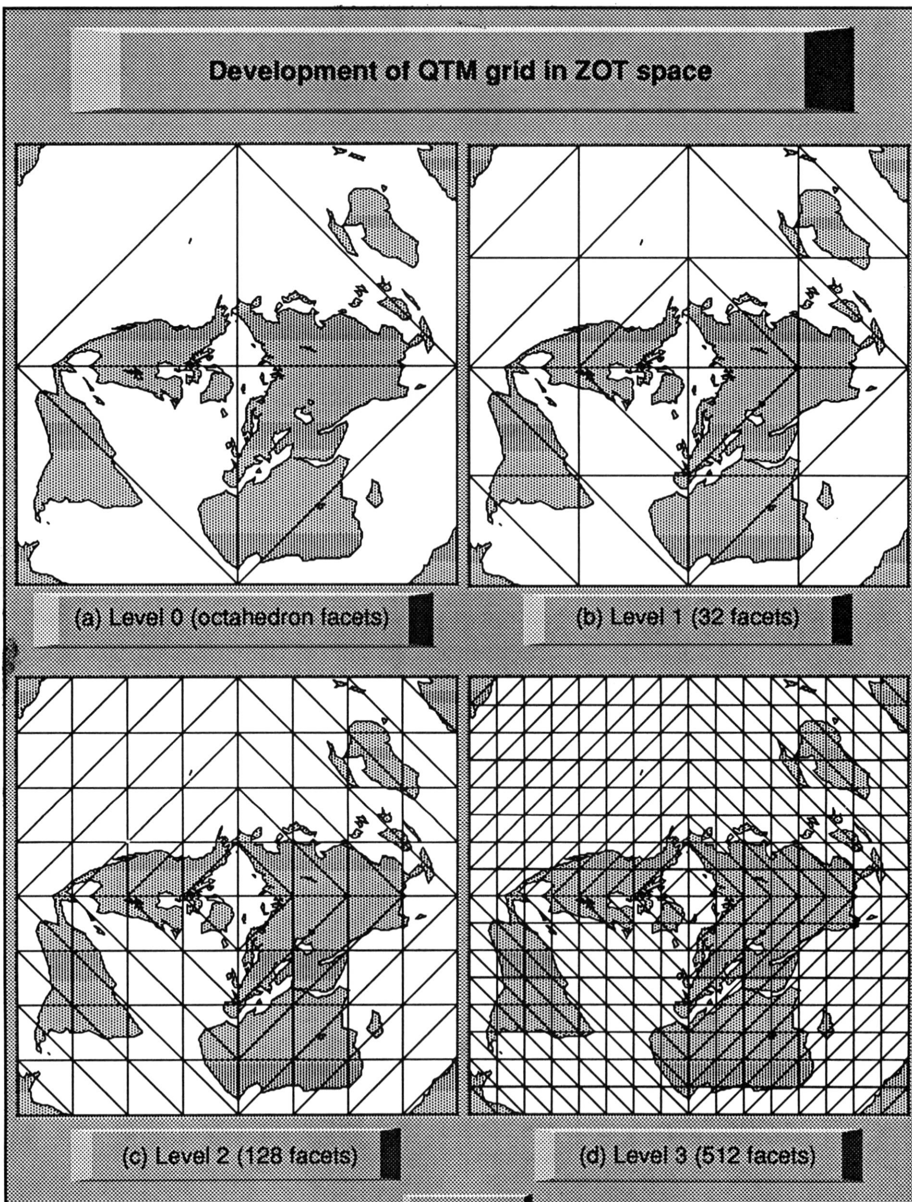

Development of QTM grid in ZOT space
(a) Level 0 (octahedron facets)
(b) Level 1 (32 facets)
(c) Level 2 (128 facets)
(d) Level 3 (512 facets)

Figure 4

with no node, but their vertices (and subsequent cells that surround them) themselves have node identifiers.

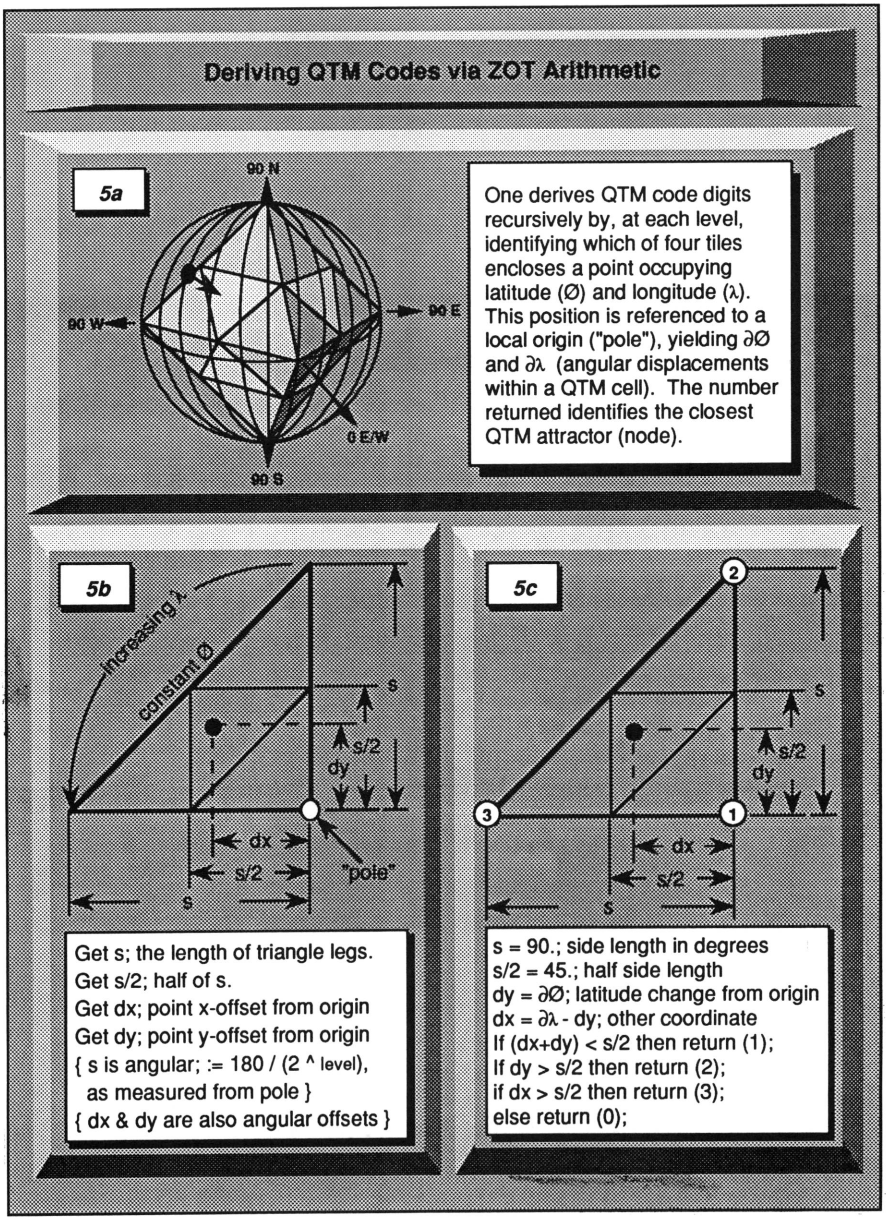
To map geographic coordinates to *QTM* identifiers, an additional procedure is therefore needed: one which identifies the "pole node"[5] (the right-angled vertex) of each *QTM* cell, and also assigns correct basis numbers to all three nodes (pole nodes can have IDs of 1, 2 or 3). This is a property not of the *ZOT* projection itself, but of the sequencing of 1- 2- and 3-cells at each level in the tessellation, which may be done as specified here, as Goodchild and Yang (1989) describe,[6] or in some other way. Another aspect of navigating *QTM* which must be parametrized is the geometric orientation of principle axes with respect to the pole node of each facet, which can be either of two arrangements per octant, one for ID's 1, 2 and 3, the other involving ID's of zero. When a point occupies a central (0) facet, the facet's orientation inverts, rotating 180 degrees. This new arrangement persists until a zero ID recurs, at which point the facet shrinks by 50 percent and flips into the other orientation. The rule is: *all facets within a given octant share its orientation unless their QTM codes contain an odd number of zeros; in such cases the current x and y scale factors interchange and change sign.*

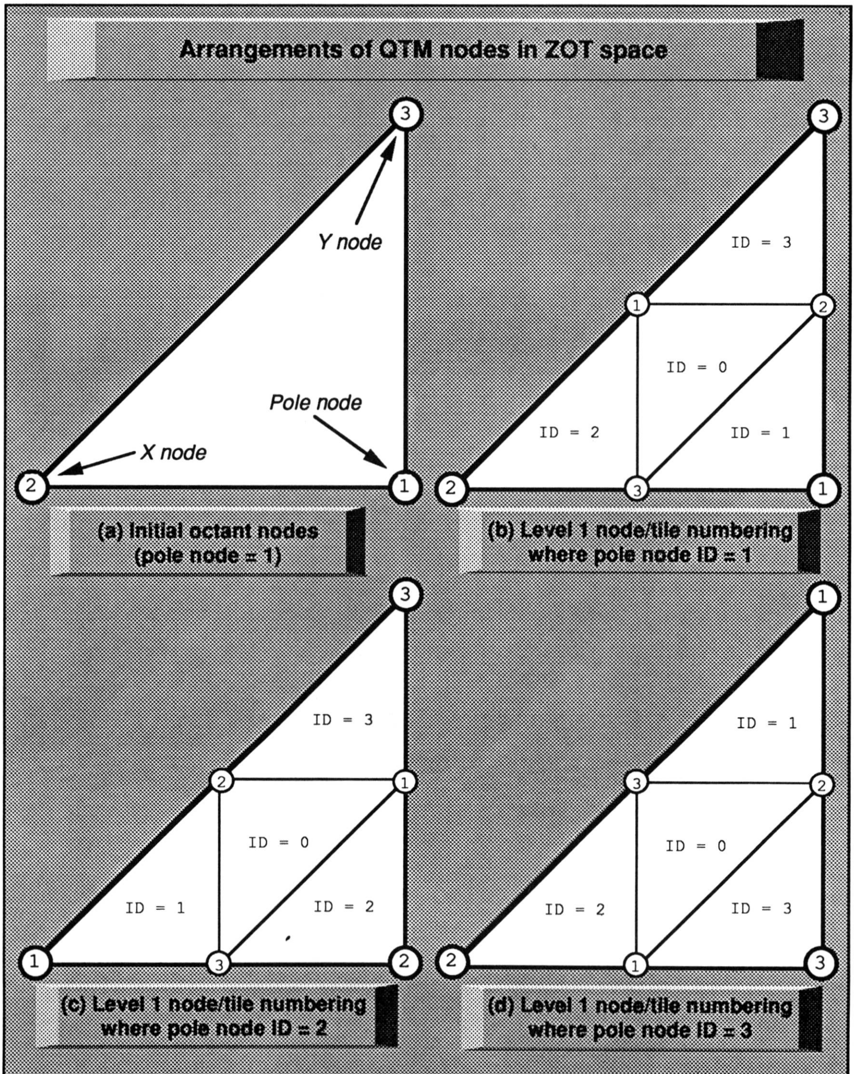
When a 0-tile comes into being, its pole node is a reflection of, and has the same ID as its parent *QTM* facet's pole node. What had been half of its parent's x-extent becomes the 0-tile's y-extent, and *vice versa*. In cases where the child tile is in the triangle dominated by the parent's pole node, its ID will be the same as its parent's. In either of the remaining two (nonzero) cases, the ID of the child's pole node flips from that of the node to which it is closest to that of the other non-pole node. Once embedded in the *ZOT* plane, transitioning to certain *QTM* ID's involves horizontal displacement, while vertical movement is used to reach others (x and y in *ZOT* space; see ***Figure 5***). Three of the six possible arrangements of nodes within an octant are enumerated in ***Table 1*** and diagramed in ***Figure 6***.

---

[5] This is the local origin of each facet, the vertex in the *QTM* mesh that, as projected via *ZOT*, has edges that all meet at right angles. Local *ZOT* distances are measured with respect to this origin, which moves each time a *QTM* ID assumes a new value.

[6] Goodchild and Yang number the tiles their mesh from 0 to 3 in one of two patterns that spiral out from the the central (0) tile first either North or South (1), then Southwest or Northwest (2), then East (3). While this scheme may simplify trilocation (generating tile IDs), it lacks one important property: There is no correspondence between tile ID's and vertex basis numbers; this makes it more difficult to relate tiles to the nodes they surround (their *QTM Attractors*).


Deriving QTM Codes via ZOT Arithmetic
5a
90 N
90 W
90 E
0 E/W
90 S
One derives QTM code digits recursively by, at each level, identifying which of four tiles encloses a point occupying latitude (Ø) and longitude (λ). This position is referenced to a local origin ("pole"), yielding ∂Ø and ∂λ (angular displacements within a QTM cell). The number returned identifies the closest QTM attractor (node).
5b
increasing λ
constant Ø
s
s/2
dy
dx
s/2
s
"pole"
Get s; the length of triangle legs.
Get s/2; half of s.
Get dx; point x-offset from origin
Get dy; point y-offset from origin
{ s is angular; := 180 / (2 ^ level), as measured from pole }
{ dx & dy are also angular offsets }
5c
2
3
1
s
s/2
dy
dx
s/2
s
s = 90.; side length in degrees
s/2 = 45.; half side length
dy = ∂Ø; latitude change from origin
dx = ∂λ - 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);



Arrangements of QTM nodes in ZOT space
Y node
Pole node
X node
3
2
1
(a) Initial octant nodes (pole node = 1)
ID = 3
ID = 0
ID = 2
ID = 1
(b) Level 1 node/tile numbering where pole node ID = 1
ID = 3
ID = 0
ID = 1
ID = 2
(c) Level 1 node/tile numbering where pole node ID = 2
ID = 1
ID = 0
ID = 2
ID = 3
(d) Level 1 node/tile numbering where pole node ID = 3

figure 6

Table 1

Basis numbers of nodes of children of an octa facet
(3 of 6 orientations)

OTM Tile	Pole	X-Node	Y-Node	- Figure 6
Parent	1	2	3	
0	1	2	3	
1	1	3	2	
2	3	2	1	
3	2	1	3	

OTM Tile	Pole	X-Node	Y-Node	- Figure 6c
Parent	2	1	3	
0	2	1	3	
1	3	1	2	
2	2	3	1	
3	1	2	3	

OTM Tile	Pole	X-Node	Y-Node	- Figure 6d
Parent	3	2	1	
0	3	2	1	
1	2	3	1	
2	1	2	3	
3	3	1	2	

Note how in each case, if a point lies nearest the parent's pole node, the child will have the same pole, but the x-node and the y-nodes interchange ID's.

Computational Properties

Because planar geometrics are generally much more straightforward than spherical ones, it is almost always easier to compute relations such as distances, azimuths and polygon containment on the plane rather than on the sphere. The former may involve square roots and occasional trig functions, but rarely to the degree demanded by geographic coordinates, where spherical trigonometry must be used no matter what ranges may be involved (unless approximations will suffice). Polyhedral geometry, being closed and faceted, is globally spherical but locally planar. The maximum practical extent of localities varies, both in cartesian and faceted cases, according to the projection employed (for cartesian coordinates) or the type and level of breakdown (for hierarchical polyhedral tessellations).

One essential operation that *ZOT* can facilitate is computing polyhedral facet addresses (geocodes) from geographic coordinates. Called *trilocation* (Dutton, 1984), it recursively identifies the ID's of tiles containing a given location, generating a sequence of L 2-bit codes, where L is the depth of recursion. The simplest general algorithm for trilocating a point in *QTM* determines which of four tiles it is in by comparing squared distance from the specified point to the centroids of the central *QTM* tile and each of the three outer ones to find the closest one; this requires 1 to 3 squared distance computations and comparisons per level, or **O** ($2L$) comparisons per point. If performed in global space, great circle distances are needed, but in the plane cartesian distances will suffice (in neither case need square roots be extracted, as we need only order distances, not measure their absolute magnitudes). *In ZOT space, computing a QTM ID requires only one addition, one subtraction, and one, two or three tests of inequality,* as demonstrated in ***Figure 5***.

ZOT casts trilocation into a well-defined planar geometry where triangular cells can be efficiently identified. Moreover, one may compute facet ID's to 15 levels of detail using coordinates stored as 32-bit integers (attempting greater precision would cause overflows and aliasing of IDs beyond the 15th level). Projecting candidate points from longitude and latitude into *ZOT* coordinates only involves solving several linear equations per point. *ZOT* distances order themselves the same as geodesic distances, and as just described, are much easier to compute.

Orientation Options

The *ZOT* projection has been shown in a specific orientation throughout this paper. As mentioned above, it is trivial to rotate the Prime Meridian to cross any point on the equator. This relocates four *QTM* cardinal points and all octant boundaries; one may be tempted to do so to avoid spreading areas of interest over more than one or two octants. Such schemes are always to the advantage of certain territories at the expense of others. Such suboptimizations are probably self-defeating, and in any case violate the spirit of the model: *QTM* can best identify locations on a planet if its mesh is embedded in a particular manifold (topological reference surface) in an agreed-upon way. Differently-oriented manifolds generate different *QTM* codes for the same location; this complicates spatial analysis, as codes from *QTM* model variants that do not share a common orientation are not commensurate, even when they represent identical locations.

QTM isn't very useful unless it is standardized, as are latitude and longitude. If nothing else, *QTM* is a coordinate system, designed to recursively encode (at some specified precision) locations on planets into unique triangular facets. It is therefore desirable that all *QTM* codes having a given address map to the same location on a planet, no matter who specified the address, where they came from or for what purpose. This implies that certain areas will always be inconveniently split by octant boundaries. Such situations can be handled by methods which knit facets together along octant edges, such as associating them with *QTM attractors*[7] (which as *figure 3* shows, follow the same pattern in all eight octants). Were everyone who used the framework to agree on how to orient it, all their *QTM* codes would also agree. Little additional data (mainly an ellipsoid model) is required beyond a common definition of the octahedron's orientation to the planet concerned.

Table 2 proposes a standard way to orient *QTM* to *ZOT*, used in illustrating this essay. It is defined by three parameters that relate *QTM* nodes to *ZOT* space: (1) The projection's *aspect* (North polar); (2) the longitudinal *offset*, if any, for the prime meridian (0º); (3) the cardinal *direction* from the central axis along which the prime meridian runs (-Y). If the geographic North and South poles are assigned ID's of 1, and the intersection of the equator with longitude 0º and 180º are labeled 2, the remaining two octahedral nodes (where the equator and longitudes -90º and 90º cross) therefore have ID's of 3. This fully defines the basis number of every node in the entire *QTM* hierarchy. The *ZOT* coordinates for x and y nodes are given in terms of the map radius (which is the length of octahedral edges as projected). These are either zero, or plus or minus unity.

[7] *QTM* nodes are also called *attractors* because all coordinates in the vicinity of a node alias to it, hence can be thought of as being attracted to that location. All *QTM* nodes beyond the original six octahedral vertices propagate their ID to six surrounding tiles, and all coordinates falling within those tiles are associated with the attracting node.

Table 2a

Proposed QTM Orientation Standard for Octa Vertices

(Octa vertices define 3 orthogonal axes upon which all QTM codes are based)

Latitude	Longitude	Pole	ZOT-X	ZOT-Y
90 N	(0)	1	0	0
90 S	(0)	1	±1	±1
0 N/S	0 E/W	2	0	1
0 N/S	180 E/W	2	0	-1
0 N/S	90 E	3	1	0
0 N/S	90 W	3	-1	0

Table 2b

Proposed QTM Orientation Standard for Octa Facets

(-x = left; +x right; -y up; +y down w.r.t. Pole node. Signs are descriptive only; node IDs are positive)

Octant	N/S	Pole	X-ID	Y-ID
1	N	1	3	2
2	N	1	3	-2
3	N	1	-3	-2
4	N	1	-3	2
5	S	1	-2	-3
6	S	1	-2	3
7	S	1	2	3
8	S	1	2	-3

Projected Implications

It is not foreseen that zenithial orthotriangular projection will ever be widely employed in published maps. *ZOT* is too peculiar to serve as an aid to navigation or to be used to convey thematic data (unless its double periodicity can be exploited)[8]. What it offers, however, is a computational shortcut for spatially indexing locations on a planet. This approach follows the lead of Lucas (1979), Diaz and Bell (1986) and others in attempting to define special arithmetics for tessellated spatial data in order to take advantage of properties of particular tessellations. Although the spaces in which most such arithmetics operate cannot be visualized as readily as *ZOT* space can, tessellar methods can have considerably higher computational efficiencies than standard geometric calculations.

[8] One might convey bivariate (or even trivariate) attribute data using a tiling of ZOT maps (as Figure 2 shows). For example, a thematic variate, such as population densities, could be displayed in a grid of M maps across, each column representing a different date in history (e.g., 1950, 1970 and 1990); each of N rows of the grid might display a different spatial resolution (one could display densities computed over the areas each nation, province or canton, one row for each scale).

ZOT can greatly simplify repetitive geometric operations in a quaternary triangular mesh, as we have tried to describe. *QTM* facets are optimally arrayed in *ZOT* space, and their addresses are highly tractable to compute. Deriving *QTM* ID's from geographic coordinates via *ZOT* is algorithmically inexpensive, growing more or less as **O**(L log$L$). So, *ZOT* may prove to be a useful cartographic abstraction, at least to the extent that *QTM* is a felicitous framework for spatial data.

References

Diaz, B. and S. Bell, 1986; *Spatial data processing using tesseral methods,* **Proc. Tesseral Workshops** 1 and 2 (1984 and 1986). U. of Reading, UK: NERC Unit for Thematic Information Systems.

Dutton, G.H., 1984; *Geodesic modelling of planetary relief,* **Cartographica**. Monograph 32-33, vol. 21, no. 2&3, ps. 188-207.

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

Eisele, C., 1963; *Charles S. Pierce and the problem of map-projection;* **Proc. Amer. Philo. Soc.**, vol. 107. no. 4, ps. 299-307.

Fisher, I., 1942; *A world map on a regular icosahedron by gnomic projection,* **Geographical Review,** October, pp. 605-619.

Fisher, I. and O.M. Miller, 1944; **World maps and globes.** New York: Essential Books.

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

Goodchild, M. and S. Yang, 1989; *A hierarchical spatial data structure for global geographic information systems.* Santa Barbara, CA: **NCGIA tech. paper 89-5**.

Life, March 1, 1943; article about Buckminster Fuller's *Dymaxion Globe.*

Lucas, D., 1979; *A Multiplication in N-space.* **Proc. Amer. Math. Soc.** Vol 74, no. 1.