GEODESIC MODELLING OF PLANETARY RELIEF

GEOFFREY DUTTON, *Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University, Cambridge, Massachusetts, USA*

SUMMARY

A METHOD for assembling and managing global terrain data is presented, the Geodesic Elevation Model. Derived from concepts in geometry, geography, geodesy, applied mathematics and computer science, GEM is designed to digitally archive and access measurements of points given in latitude, longitude and elevation from any source, by embedding them in a regular, polyhedral data structure. To do this, the model recursively tessellates a regular solid, initially an octahedron, into equilateral triangular facets. Spot measurements are encoded by successive approximation, mapping a given geodetic location to proximal centroids of nested triangles. As encoding proceeds, a new vertex appears at the center of each existing facet; an elevation code for it is entered in a linear tree, an estimated coordinate which locally wrinkles the polyhedron. The more times this takes place, the better is the approximation: each such step of encoding triples the number of facets, and diminishes horizontal and vertical error by the square root of three. As the structure is a regular geodesic grid, its horizontal coordinates are implicit by their ordering. Elevations alone are stored, using 1-bit flags quantizing height changes, triangulating the enclosed terrain with less than one bit of data per facet. Reconstruction of the data yields estimates of longitude, latitude and elevation anywhere on the planet, along with the error of estimate. Consequently, the slope, size and aspect of facets can be derived at any level of precision required, up to the limit of detail encoded for their neighborhoods. Beyond this, if desired, fictitious detail can be fractally synthesized, landforms resembling features above them in the hierarchy, smoothing the surface simultaneously. Local regions (small initial facets) can be encoded and stored independently, then subsequently merged at will to assemble larger terrain models. As measurements accumulate in a GEM database, superfluous and erroneous data are rejected with increasing frequency, due to the self-calibrating nature of the ensemble. Were sources of data and motivation sufficient, the relief of the entire Earth could be uniformly encoded in GEM format at a horizontal resolution of less than one kilometer, with a vertical precision of several meters, on a single disk volume.

FIRST

The study of the Earth and its spatial properties now encompasses many disciplines. Those stemming from roots in classical geography include geodesy, surveying, cartography, geomorphology and hydrology, now augmented by photogrammetry and remote sensing. Despite their common origins and subject matter, they have diverged in their focus, methodologies, literature and empirical content. Nevertheless, they remain necessary adjuncts to one another in describing the shape of our planet and the relief of its surface.

Despite persisting political rhetoric emphasizing global territorial, ecological, economic and ideological divisions, we live on one planet, a world which may be regarded as an organic entity. The wholeness of the Earth – self-evident to many non-western cultures – largely escapes the western culture of science, even the so-called earth sciences. This peculiar myopia is reflected both by our disciplinary specialization and by territorial concerns.

This is not to assert that science is ignorant or unmindful of the form of our planet. Were this the case, military frontiers would still be boundaries on maps instead of trajectories between points over the horizon. The imperatives of strategic warfare have both demanded and generated accurate models of the shape of the Earth. Digital methods, aerospace technology and geodetic models now enable strategists to calculate distances between launchers and targets half a world away with deadly precision. But if asked to identify which major cities would suffer inundation should melting icecaps cause the oceans to swell by one percent, few earth scientists could do more than guess or crudely approximate. There are too many floodplains and too little certainty of their shapes.

The aforementioned scenario could develop over the next century or so, and its probability may have been increased by the consequences of human activity, amplified through technology. Ironically, it is only through using technology that we have any hope of comprehending the nature of the problems we have wrought for the planet. But assembling relevant data in a holistic, globally useful framework – a seemingly obvious necessity – is rarely attempted, and such efforts are often frustrated by parochialism, inconsistencies and errors, not to mention inadequate models of the structure of the planet and the processes at work within, on and above its surface.

Digital Terrain Modelling

Identifying the extent of floodplains is just one of many cartographic and analytical tasks facilitated by digital terrain models (DTMs). Numeric representations of surfaces are becoming increasingly important in civil engineering, regional analysis, military operations and topographic mapping. Over the past fifteen years or so, a number of data structures and formats have been developed to encode topographic surfaces (Mark 1977), some of which are exhaustive enumerations of elevations throughout a region (or triangulations of 'surface-specific' points), others of which are mathematical (usually coefficients of polynomial or trigonometric series, approximating relief as smooth surfaces).

Perhaps the purest example of a mathematical terrain model, and in many ways closest in spirit to the model presented in this report, was formulated between 1908 and 1922 by Prey (1968). In that model, the shape of the earth was represented by spherical harmonic equations to order sixteen. To have refined his model further would have exceeded the computational power and demanded better data than were available in Prey's time.

Since then, not only has the development of computing technology vastly increased our command of techniques, this revolution has unloosed an avalanche of spatial data, burying our abilities to catalog, verify, analyze and apply it. The work reported here is, like Prey's effort, an attempt to bring some unity to the chaos of DTM technology and data.

Gridded Terrain Models

The most common way to represent geographic data is to accumulate observed or interpolated point data into fixed rectangular grids or irregular triangular meshes. The former type are by far the dominant form for DTM data, as they can be compiled nearly automatically using analytic stereoplotters to scan stereo pairs of aerial photos, correlating features to compute elevation profiles as a dense raster of points. DTMs produced by such machines (after resampling and editing) are publicly distributed by the U.S. Geological Survey; these are available for a growing number of 7.5-minute quadrangles, representing their terrain with about 170,000 grid points spaced at 30-meter intervals. Quality control of such data is problematic, as the accuracy of the output from the analytic stereoplotters varies with terrain type and land cover, and in particular is affected by the presence of water bodies and man-made structures.

Triangulated Irregular Networks

The principal alternative to rectangular arrays for storing DTM data is the Triangular Irregular Network (Peucker et al 1977; Males, 1978; Gold 1978), or TIN. In this model, elevations are digitized from maps manually, selected to represent critical features of the surfaces to be encoded, such as peaks, pits, passes and breaks in slope. These points are then triangulated, either manually or analytically (Fowler and Little 1979), yielding a varying network of triangular facets fitting the terrain, containing most of the information about the surface with relatively little input data. While some thought must be given to selecting and connecting the initial spot elevations, the resulting model is more compact and useful than gridded DTMs, principally because of the properties of triangles and the networks they form. The main drawback to TINs is the complexity of the data structures and programming strategies needed to manage and apply them. Furthermore, while all implementations of TINs are conceptually equivalent, their structures differ sufficiently to make data transfer from one system to another difficult. Transferability issues also arise for gridded DTMs, but usually due to differing data formats (ordering and character encoding conventions) rather than because of any fundamental differences in data structure.

Accuracy Issues

Despite their different constructions, there are some common aspects to and shortcomings of the two models just outlined. Their principal similarity is that both grids and TINs are designed to encode planar coordinates for relatively small areas. That is, the horizontal coordinates (which are explicit in TINs but implicit in grids) are almost always cartesian, and scaled to whatever map projection was used to compile the source maps or photographs from which they are digitized. In too many cases, the projections are not documented, leading to difficulty when adjacent terrain models are merged or when particular ones are modified.

Another set of limitations relate to the accuracy of DTM data. Gridded data have an implicit limit to horizontal precision (its Nyquist frequency), which is nominally uniform throughout the grid. In practice, however, gridded surfaces may be produced by interpolation procedures from scattered observations. The resultant precision is thus variable, but its magnitudes are hidden, unless one has access to the source data.

Vertical precision, likewise, may or may not be uniform throughout a grid, depending on the methods and sources used to compile it. Moreover, a structural interdependence exists between it and horizontal precision (sampling density), demonstrated in Dutton (1983). If grid cells are large they may contain considerable amounts of height variation, so that the value assigned to each is only an estimate (high, low or average) of heights within the cell. To ascribe high precision to the elevations assigned to grossly-sampled cells is thus rather pointless unless the terrain is generally smooth. Therefore, the amount of memory needed to represent the height of a grid cell grows larger as sampling density increases. Specifically, the number of bits needed to encode each cell is proportional to the logarithm of the number of cells.

Precision in TINs in subject to similar constraints, as each control point may vary in how well it represents conditions in its neighborhood. In general, precision will vary inversely with the spacing of observations, but need not (and probably will not) be the same at each control point, due to source errors, variations in operators' performances and triangulation decisions. While each coordinate and face in a TIN can be labelled to document its presumed precision, this is never done in practice. To do so would erase much of the storage efficiency enjoyed by TINs.

In editing gridded surfaces and TINs, gross errors can be detected automatically by algorithms which identify drastic changes in slope or linear artifacts. Visual inspection is still the best way to achieve quality control, but only errors that result in discernible patterns are likely to be rectified. The overall fidelity of the data is difficult to assess without detailed information of how the source data were collected, edited, reproduced and (sometimes) interpolated. The datasets themselves are completely indifferent to the quality of the information they contain, and this implies that they will usually contain errors which will persist without notice, but unfortunately not without consequence, indefinitely into the future.

FORM

In the following sections, a structure for encoding terrain is presented which differs significantly from both grids and TINs in both the vertical and horizontal components. Like a grid, elevations alone are explicitly encoded, in a regular

Table 1 COMPARISON OF GRID, TIN AND GEM STRUCTURE AND PROPERTIES

DTM:	GRID	TIN	GEM
BASIS	Raster	Landforms	Planet
FORM	Cartesian	Triangulated	Polyhedral
TOPOLOGY	Implicit 2-d	Explicit 2-d	Implicit 3-d
SAMPLING	Uniform	Irregular	Hierarchical
PRECISION	Fixed	Variable	Convergent
CONTENTS	Elevations	x,y,z,pointers	Diff. Codes
STRUCTURE	Array	linked lists	dual trees
STORAGE/CELL	medium	high	low
STORAGE/MAP	high	medium	low
ACCURACY	standardized	uncertain	verifiable
COMPLEXITY	low	high	medium
COVERAGE	local	regional	global
EDITING	tedious	complex	automatic

mesh; like a TIN, all cells are triangles with identifiable faces, vertices and edges. Unlike either, however, the model is designed to be planetary in scope, capable of accepting observations from any location on earth (or whatever planet it represents), storing them in geodetic (spherical) coordinates. Furthermore, if observations are properly labelled as to their horizontal and vertical precision, the model will encode each to the appropriate tolerance and no further. When they are sufficiently in error, input data can be rejected by the model automatically. These and other useful properties are achieved with surprising economy; each facet encoded requires storage of *less than one bit* of data, making the model at least an order of magnitude more compact than either grids or TINS. This method of encoding planetary relief has been named Geodesic Elevation Modelling (GEM). Some of its properties are summarized in Table 1, in comparison with gridded and triangulated data structures.

Geodesic Tesselation

A geodesic structure can be generated from a polyhedron (usually an icosahedron[1] or portion of one) by regularly subdividing it in several well-defined ways. In two of these methods, the so-called 'triacon' and 'alternate' breakdowns (Popko, 1968), a higher-frequency grid of triangles is created by connecting either the centroids and vertices of triangles (triacon, Figure 1b) or their edge midpoints (alternate, Figure 1a). The two procedures yield 60 and 80 triangular facets, respectively, from the original 20 faces of the icosahedron. Each of these new triangles can subsequently be broken down in the same manner, each time tripling or quadrupling the number of faces in the structure. Eventually, the triangles grow quite small and the figure begins to closely approximate a sphere. Ten orders of subdivision or less is about the limit for engineered structures; edge members grow quite short, yet of slightly (but critically) differing length, posing tolerance problems in manufacturing them and their connectors. Computational models, however, can be made of such structures without encountering such problems, at least to one part in several million.

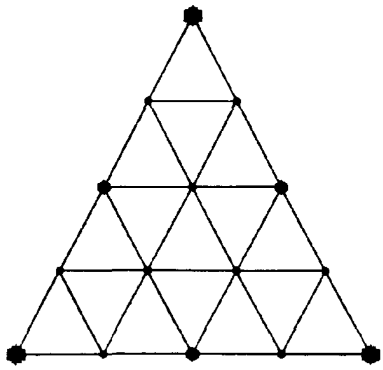


FIGURE 1a. *The 'alternate' geodesic hierarchy.*

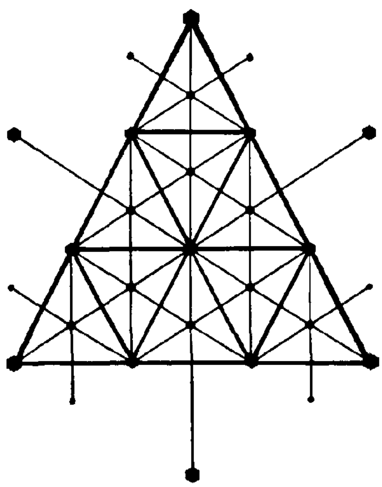


FIGURE 1b. *The 'triacon' geodesic hierarchy.*

Polyhedral Projection

While geodesic structures are normally regarded as minimal space-enclosing ones, they can also be thought of as models of surfaces which are, topologically speaking, spheres. Fuller and Sadao (1982) have published a 'globe' which represents the earth as an icosahedron, the so-called 'Dymaxion' projection. Although ingenious (it can be cut and folded from a single sheet of paper) and informative (it offers many untapped possibilities for thematic mapping), the Dymaxion projection is not widely used, being regarded by mapping professionals as a cartographic curiosity. While the Dymaxion Map distributes its projection errors predictably, its distortions are concentrated at vertices and arbitrarily located, depending on the orientation of the icosahedron with respect to the earth's axes. Where accuracy must be predictable, then, the Dymaxion projection is not the one of choice.

These limitations recede, however, as one proceeds to subdivide the icosahedron into smaller triangles. With each successive division edge lengths are reduced by half (for the alternate breakdown) or by the square root of three (triacon breakdown) and the polyhedron grows more spherical. Its facets lie closer to sea level and projection distortions relax. After only five (alternate) breakdowns, the original twenty icosa faces blossom into 20,480, with vertices arrayed 2.25 degrees apart. This rate of expansion can be continued to make the polyhedron indistinguishable from a sphere for any cartographic purpose.

Fuller uses the icosahedron as the basis for most of his geodesic structures, a choice which is both theoretically and pragmatically justified for engineering purposes. But despite the appeal of icosahedral forms for space-frame structures, they may not in fact provide an equally optimum basis for modelling geodetic relief. The facets of an icosahedron are tilted with respect to the equator and prime meridian, complicating computations of their geodetic coordinates.

An *octahedron* (having 8 triangular faces, 6 vertices and 12 edges), on the other hand, can be aligned to cardinal points, and this property leads us to regard it as a more appropriate polyhedron upon which to structure a geodesic data base. If oriented so that the polar axis passes through two opposite vertices and so that the prime meridian and the equator intersect at another vertex, the octahedral kernal acquires certain useful properties. First, all 'baselines' of the initial facets and their subdivisions are parallel to the equator. Furthermore, the difference in latitude or longitude between vertices is divided by three with every pair of subdivisions. This yields an isometric graticule which grows finer with each breakdown, three sets of 'standard parallels', two of which are oblique. Each face is semi-uniquely defined by the three parallels which intersect about it (two such faces are formed at antipodes). One potential disadvantage, however, is that the shape of facets will vary from nearly equilateral (near octahedron face centers) to right spherical triangles (at octa vertices), and consequently their areas will differ. The size and shape of any facet can always be computed, and in any case their variation does not make the model any less useful for storage and retrieval of elevation data.

FEATURES

In the following sections, one implementation of a Geodesic Elevation Model is described. This model is based on the *Triacon Breakdown of an Octahedron*, ignoring other geometries, but not dismissing them as inappropriate. Were other breakdowns of other figures to be employed, the logic of the model would be very similar; certain parameters (such as the rate at which facets multiply) and data structures (memory addressing strategies) would be altered, and certain convenient properties might be forsaken.

Like any digital terrain model, GEM has three major related elements:

1 Horizontal Organization
2 Vertical Encoding
3 Data Structure

To these might be added a fourth element, not always explicit in DTMS, error estimation and control. GEM's horizontal organization is that of the octahedral

Table 2 'TRIOCTACON' TESSELLATION OF A SPHERE WITH A RADIUS OF 4000 MILES

LEVEL	NO FACES	NO POINTS	NO EDGES	EDGE LEN	FACE AREA	VERT ERR.
1	8	6	12	6531.969	18475188.	1952.134
2	24	14	36	4046.708	7090952.	634.494
3	72	38	108	2390.866	2475205.	211.091
4	216	110	324	1390.952	837770.	70.350
5	648	326	972	805.109	280679.	23.449
6	1944	974	2916	465.224	93718.	7.817
7	5832	2918	8748	268.673	31257.	2.605
8	17496	8750	26244	155.133	10421.	0.869
9	52488	26246	78732	89.569	3474.	0.290
10	157464	78734	236196	51.713	1158.	0.097
11	472392	236198	708588	29.857	386.	0.032
12	1417176	708590	2125764	17.238	129.	0.011

triacon breakdown, a regular tessellation which, like a raster-encoded image, needs no horizontal coordinates: location is implied by position in the data structure. The two alternating hierarchies of the triacon provide a triangular matrix of control points regularly arrayed across the surface of the planet. Consequently, only vertical information need be contained in a GEM file, encoded as bit-flags which signal the elevation change at each control point. The flags describe the direction of elevation change, but not its magnitude. The amount of vertical movement is, analogously to horizontal offsets, given by the position (depth) of facets in the hierarchy. This relatively unexplored method of elevation modelling has been named DEPTH, for 'Difference-Encoded Polynomial Terrain Hierarchy' (Dutton, 1983). Its effect is to approximate elevations vertically to a similar extent that locations are approximated by triangular breakdowns. Each level of a DEPTH hierarchy consequently encodes more control points with greater precision than the levels above it. Table 2 enumerates this hierarchical progression to twelve levels, illustrating the asymptotic approximation of a sphere of Earth radius starting from an octahedron fitted around it. This series of breakdowns converges rapidly. Its faces multiply by powers of three, reducing the triangles at the 12th level to less than 130 square miles apiece. By this level, vertical error is such that the center of each triangle is about 50 feet from the spherical surface. As implemented in GEM, however, the rate of spherical convergence is reduced from a ratio of three to the square root of three.

GEM's archival and working data structure represents vertices as two parallel sequences of levels in dual nonary trees. In each hierarchy every non-terminal triangle divides into nine, each with one third the edge length of its parent. Because all descendant nodes are represented in the tree, there is no need for pointers; the tree is laid out as a series of arrays, each three times (more or less) longer than the last. Access to values is then through computing an address from the coordinates of a centroid and its depth in the tree. Similar data structures for quadtree hierarchies are in use, sometimes called 'linear trees' (Gargantini 1982). In the absence of list pointers, all operations on such structures proceed sequentially, top-down.

Horizontal Encoding

GEM's triacon grid has the useful property that centroids of triangles mark elevation nodes, which then serve as vertices for triangles in the next lower level. Every terrestrial location can be approximated within *epsilon* distance units by a specific sequence of triangulations converging about it. The primary facet can be one of the initial polyhedron, or one of its divided facets. In either case, the same process is used to continue breakdown. The procedure generates a series of partially or completely nested triangles, the numbering of which provides both a geocode and a key for memory addressing, as well as the vertex coordinates of each triangle in turn.

To characterize the recursive strategy of approximating locations via successive triangulations, the term *trilocation* has been coined. Use of this neologism will simplify subsequent discussions, as should its syntactic variations, such as *trilocate, trilocated* and *trilocal*. Computations for trilocating can be very simple, due to the regularity of the geodesic structure. Algorithmically, to trilocate point [pq], given that it is bounded by vertices [p1,p2,p3], perform:

```
PROCEDURE TRILOC (Real: pq,p1,p2,p3,p1new,p2new,p3new,center,epsilon)
; Converges a triangle around a point to a new one one-third as big;
; Vertices are pairs of planar or spherical coordinates.
; pq ::= point to be located; (p1,p2,p3) ::= current vertices;
; (p1new,p2new,p3new) ::= returned convergent vertices;
; center ::= centroid of new triangle; epsilon ::= error tolerance;
  BEGIN
    Determine squared distance from [p1] to [p2]
    If less than [epsilon] squared then return; we're close enough.
    Determine squared distances from [pq] to [p1], [p2] and [p3]
    Sort them, ordering points as [pnear], [pmid] and [pfar]
    Compute vertices of convergent triangle containing [pq]:
      [p1new] ::= [p1 + p2 + p3]/3; centroid of current triad
      [p2new] ::= [pnear]; closest old vertex
      [p3new] ::= [pnear] + [pmid] - [p1new]; centroid of nearest neighbor
      [center] ::= [p1new + p2new + p3new]/3; best estimate of all
  END.
```

Each time TRILOC is called, the edge length of resultant triangles grows smaller by the square root of three. This creates an alternating sequence of triangles, in which every level is fully contained within the level two steps above it, and has nine times as many triangles, each with one-third the edge length. This pattern of breakdown is shown in Figure 1b, with odd-numbered breakdowns drawn with bold lines. The trilocation procedure is diagrammed in Figure 2, showing the approximation of a location in six steps. Note that while each trilocation generates bounding vertices, it is the center of that triangle which marks where vertical information is encoded at that level. Figure 3 illustrates the initial polyhedral form of the GEM 'trioctacon' (triacon breakdown of an octahedron) structure. Compare it with Figure 2 to imagine how trilocation generates an alternating sequence of subfacets from the triangles of the trioctacon.

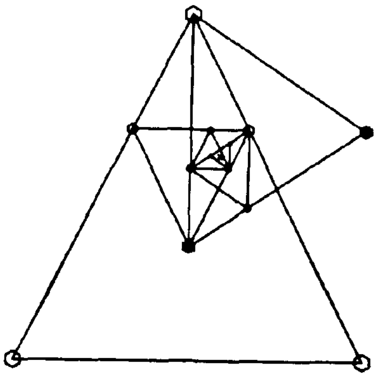
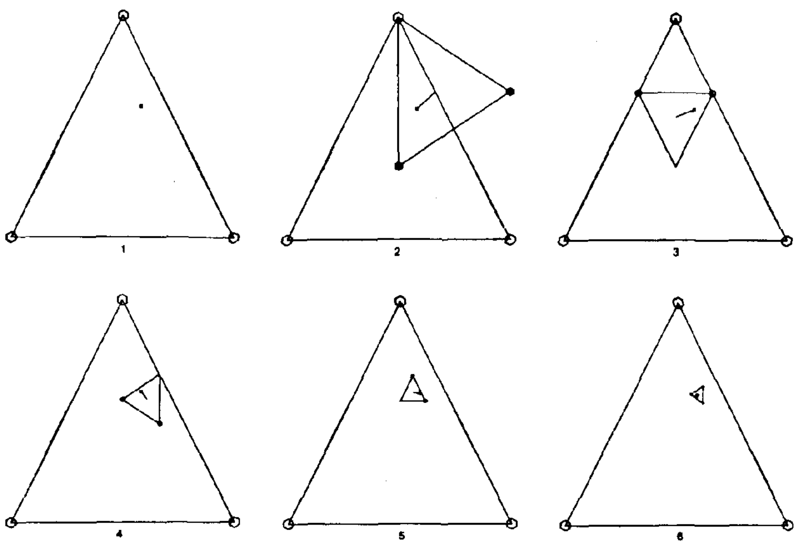


FIGURE 2. *Geodesic spatial search: the trilocation procedure. The small cross marks a location to be found. Six trilocations are shown, converging to within 4% of initial edge length. The 'best' estimate at each stage is the center of the current triangle.*



Vertical Encoding

Considered three-dimensionally, the two alternating triacon grids can be regarded as constituting a pair of concentric polyhedra. The odd levels, those of the initial octahedron and its subdivisions, generate an object which shrinks slightly with each subdivision. Those of the other form start as the dual of the octahedron, a cube. As this network is subdivided the volume it contains grows larger. If the radius of the octahedron is initially set to be somewhat larger than the earth (actually, 7.75 percent larger), the radius of the cube (where the radius is the distance from its center to any vertex) will be smaller than Earth's by the same percent. In the process of subdividing, the vertices of both figures tend to converge to Earth radius, eventually approximating spheres which are almost coincidental.[2]

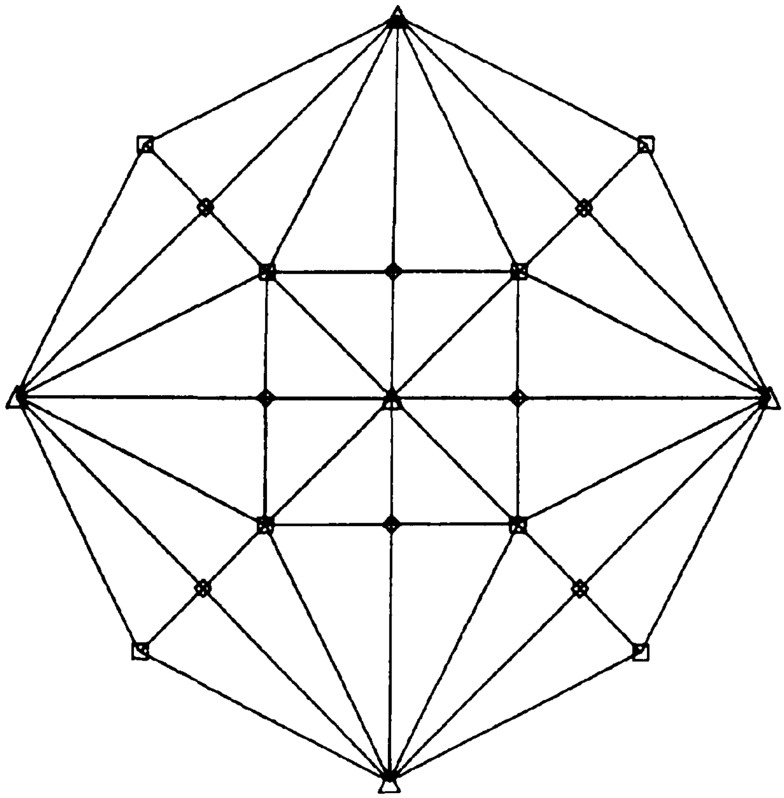


FIGURE 3. *Schematic of the initial* GEM *'trioctacon' lattice: octahedral vertices are marked with triangles, cubic ones with squares. Intersections of the networks are marked with diamonds. This is an 'inverted' perspective drawn as if viewed from a nearby vanishing point. Note the diamond-shaped facets which connect cubic and octa vertices. These are the faces of a Rhombic Dodecahedron, a form which contains both initial polyhedra.*

What DEPTH encoding does is to add information to each vertex in each level of both hierarchies which causes this process of convergence to depart from sphericity. Flagging on a DEPTH code for a vertex can cause it to locally 'dimple' or 'pimple'. Because two networks are involved, only one kind of wrinkling occurs at each. That is, the initial (octahedral) network, being larger in radius than the Earth, will contain the pimples (peaks and ridges), while the cube-based network, smaller than the Earth initially, will encode the dimples (pits and courses). Each vertex is assigned a single bit of storage, and will wrinkle if that bit is set, but will maintain a smooth curvature if its bit is not set. The direction of the wrinkling is fixed for each network (which is why only one bit is needed to encode it). The magnitude of the wrinkling is constant for all vertices at a given level, and diminishes geometrically (by root three) as triangles grow smaller, the dual networks converging.

DEPTH represents (discrete) *changes* in elevation rather than (continuous) heights. It works by comparing the value of an elevation to be encoded with the prior estimate for it (which can be arbitrary initially); if the difference between the actual and the estimated heights is greater than the tolerance in effect for that level, a flag is set to indicate that the estimate is to be raised or lowered by the current step size (which will be related to the tolerance). Moving down the hierarchy, step size gets smaller, halting the encoding when the displacement becomes less than some stated vertical error tolerance; this reflects either how precise the source data is presumed to be or how accurately one wishes to encode it.

Formally, this constitutes a polynomial series (Dutton 1983, eq. 5) for each vertex, generating a weight for each term as a dummy variable. The weights are then multiplied by the step size for their level. Terms with weights of unity are added (if even) or subtracted (if odd); if a weight is zero, its term is ignored and displacement is toward sphericity. GEM's data structure is simply a way of ordering these two series of weights, bit after bit.

Elevation flags can be grouped into files containing fields for a specified number of levels. It is probably advantageous (and certainly simpler) to make the hierarchical depth (hence size and layout) of each file identical. Within a file, vertices can be clustered by level and arrayed contiguously, ordered by vertex number. This number, in essence a mathematical geocode, is assigned by counting vertices in a uniform way at each level. When finding the central point of a facet via trilocation, its location code can also be computed, then converted into a (byte) memory address for the linear tree. To be most useful, the vertex numbering scheme should generate unique geocodes directly from latitude, longitude and level. The numbering problem has not been fully solved for GEM, and suggestions in this regard are welcomed.

Programmability Issues. Despite that (or because) organizing and manipulating GEM data involve such primitive operations, not every programming language makes them easy. As data are arrayed by bits and addressed by bytes, using languages such as FORTRAN, BASIC and even PASCAL invites hacking byzantine, inefficient code. More polymorphous ones, including C, FORTH, LISP and of course Assembly Code, support GEM operations much more efficiently. In the current research effort a hybrid approach is being attempted, using C to code primitive operations and FORTRAN for higher-level procedures. In our environment (Vax 11/780, VMS), this is accomplished easily, without recourse to Macro coding.

Quality Control

At one time every introductory lecture on data processing seemed obliged to note the acronym GIGO: Garbage In, Garbage Out. This rubric is usually a preface to the assertion that computers are mechanical, obedient and stupid, their software incapable of evaluating the integrity of information being manipulated. For most applications programs and database systems, this still is more or less the case. While each variable and parameter may have a defined set or range of valid values, incorrect ones can still penetrate such coarse filters, biasing tabulations and statistics, polluting databases.

As sets of measurements go, spatial data are distinctly corruptible. Map data have at least three dimensions (spatial location and value), each of which is subject to several sorts of error at various stages of compilation. The passage of time tends to blur the significance of most measurements, especially those that are catalogued without citation of sources and estimates of accuracy. Moreover, it is often more trouble than it is worth to attempt to overlay or merge map files compiled at different times, scales and in different formats, seriously hampering their utility for analysis and planning applications.

Validation. Despite having a simple structure and lacking redundancy, a GEM database has considerable capacity to detect and reject suspicious data, provided that it is initialized with accurate measurements which are fairly widely distributed, rather than being clustered in a few locations. Furthermore, all coordinates input must be explicitly tagged with estimates of horizontal and vertical accuracy to be acceptable. These error terms then determine the lowest level in the structure to which encoding can proceed. At some stage, vertical error will exceed the magnitude of vertical displacement, making it no longer possible to confidently assign a DEPTH code. Similarly, once horizontal error exceeds about one-half the distance between neighboring vertices, it is no longer possible to locate an observation in a particular facet. This means that less certain measurements are limited to the higher reaches of the hierarchy, while more precise ones dive deeper into it.

Tuning. Once installed in the tree, a spot measurement fixes the heights of all the vertices it has visited during its trilocation. Should another measurement subsequently be inserted which follows the same path (for any number of levels), because it is located near the first one, it must generate DEPTH codes which agree with those already in place. If, during this process, a DEPTH code is generated which conflicts with one already stored for a facet, the accuracy of one or both measurements is called into question. A general rule for resolving such contentions can be that the measurement having the smaller error terms (hence a greater depth of encoding) wins. Normally, one tries to insert the most reliable data into a GEM database first, so that they can referee subsequent measurements in their neighborhoods. For example, one could pin down the structure by first encoding known survey monuments, mountain peaks nd coastlines (where tidal inundation is not extreme) before entering less verifiable elevations. If, however, a subsequent measurement has greater precision than one already encoded, it should logically supersede it. This can be done, but may require a number of values inferior to the conflicting vertex to be modified in the process.

Adjustment. To verify whether a candidate elevation is more precise than any measurement already stored within a facet where conflict occurs, one searches for set elevation flags inferior to the facet. If none are found, the value can be entered without altering any other data. Otherwise, flags will have to be re-set to maintain the heights currently encoded for them. A refinement is to adjust elevation codes outward from the node just encoded, interpolating values for neighbors at levels between their current depth and that of the new measurement.

Interpolation. The final triangulation estimates elevations at the places where the last two levels cross (at the middle of their edges). It is only at these locations that the two hierarchies meet. All other vertices are either in one network or another. The height computed will be the average of the four vertices which define the intersection. The result generates a network of right triangles defining the surface of the planet. Two such triangles are generated for each one in the lowest level encoded, doubling spatial resolution.

FACETS

Like any array storage structure, GEM provides a place for everything, with everything in its (approximate) place. As GEM storage is hierarchical, the size of the place a thing occupies can vary with its depth. Space must be pre-allocated, both in memory and in files, for all vertices throughout a breakdown, for as many levels as one intends to encode. While only one bit per vertex is needed, the number of them grows geometrically. Specifically, in a hierarchy of N levels of triangles, there will be 3**(N-1) ultimate vertices (leaves in the tree), and (3**N)/2 vertices overall, requiring (3**N)/16 bytes to fully represent the structure. To give examples, six levels will require just 46 bytes of data space, but 12 levels demand 33,216 bytes, and to hold 18 levels 24,213,781 bytes must be committed.

Reserving storage is cheap to begin with, but clearly grows prohibitively expensive beyond 12 levels or so. One thus must accept a tradeoff between a unified, inflated structure and a limited, manageable one. At the expense of slightly greater algorithmic complexity and processing time, the geodesic hierarchy can be segmented into two or more orders, avoiding the inefficiency of storing unencoded areas. This means that each initial facet is rooted in a first-order file which contains L levels of detail. Another L levels of further detail are then available from a group of second-order files: as many of these may exist as there are leaves in the first order tree, although none need be created unless elevations are encoded for the facets they contain. When encoding or searching beyond the first order, the appropriate second-order file is created or read into memory, doubling the amount of data held there.

The type, size and format chosen for GEM files depend on many factors, few of which are intrinsic to the model itself. Fast direct access is desirable, but properly buffered, sequential files can be highly efficient. For simplicity of access and update, fixed block random files may work best. As already mentioned, file size can inflate enormously if one attempts to encode many levels of the tree at once. In order to tune GEM to virtual memory environments, data should be blocked into page-sized records, such as 256, 512 or 1024 bytes. Given a 512-byte page size, it should be possible to store eight levels of triangles (more than 6500 faces) within it, accessing them in a single read instruction. This tessellates the Earth into facets roughly 3,500 square miles in area, defined by vertices spaced about 100 miles apart. Eight such pages, 4096 bytes in all, would thus represent the earth's shape as a first approximation.[3]

To then extend the hierarchy another eight levels, each of the 52,488 facets of the first order (8 sets of 6561 facets) would generate 6561 facets of its own, requiring storage of one page of data apiece. This amounts to nearly 350 million facets for the whole planet, and uses slightly less than 27 million bytes to store their DEPTH codes. Each 16th-order facet intersects with a 15th-order one in the final interpolation, yielding about 690 million ultimate triangles. Typically, they cover about 0.075 square miles in area, and have vertices spaced about 0.3 miles apart.

Geocoding Considerations

Whether or not an actual GEM database is involved, GEM's geodesic tessellation procedure can be employed to compute a hierarchical address for any point on the earth's surface. The address is simply the concatenation of facet numbers created by subdividing the original eight faces of the initial octahedron into nine facets each. The address will be shared, of course, by all other points lying within the lowest-order facet computed. Each subdivision contributes a low-order decimal digit as it decomposes the facet into nine nested ones, a rapidly converging procedure. At some level of subdivision the triangle containing a specific location will be sufficiently small to encode it within its presumed locus of error or within an acceptable distance for the purpose at hand. The sequence of triangles visited in arriving at this determination constitutes the geocode. For example, the eight-digit geocode '69825846' will isolate a point somewhere on the planet, distinguishing it from 8*(9**7), or more than 38 million other possible facets, each of which occupies about one-quarter of a square mile. Eight or nine levels of GEM geocoding would therefore normally suffice to uniquely identify land units at the scale of quarter sections, including most postal zones and census tracts.

As it happens under one possible GEM numbering system, the 'gemcode' 69825846 occupies a portion of lower Manhattan in New York City, a triangular area containing the World Trade Center, and part of Wall Street, but not large enough to include City Hall. The sequence of facets leading to this area is illustrated in Figures 4A through 4F. The figures show detailed U.S. state outlines superimposed on the sequence of GEM facets leading to the address 69825846 at six different scales. Only those facets which contribute to the trilocation of lower Manhattan are included, 64 triangles in all. The apparent discrepancies in the locations of edges shared by successive levels is due to the distortions of the orthographic projection (centered on Washington, D.C.) used for the maps. As these chords shorten and lie closer to the Earth's surface, the disparities become unnoticeable.

The relevance of this somewhat abstract framework is its capability to generate a spatially meaningful geocode for any location, and more importantly, its ability to derive standardized coordinates from any given geocode. Other hierarchical systems have been devised which have some of these properties, notably UTM. All of them, however, work in a planar domain, and are tied to specific projections which must be referenced to particular zones or map sheet origins. GEM addressing is planetary in scope, uniform in its geodetic structure, and carries with it an explicit statement of accuracy. Consequently, spatial partitioning schemes such as GEM can generate geocodes which could concretely denote coordinates. If users could agree on rules for generating GEM geocodes, it would be possible to replace all the coordinates in a cartographic data file (each of which is normally described by two 32-bit words) by a 64-bit (in two 32-bit portions) hierarchical geocode, capable of resolving points several inches apart.[4] This would then enable the extraction of the data at lower resolution for use in changing scale or equating the precision of coverage boundaries for overlay. Such precision is certainly far in excess of what would be needed for digital terrain modelling.

Note that the geocoding scheme illustrated here uses only the facets in the octahedron-based network, ignoring those in the alternate (cube-based) one. Since the two networks overlap, using both in a geocode would cause ambiguities, even though finer scale distinctions would be possible. As it is, each successive digit in the geocode provides a threefold refinement in spatial resolution, hence scale. Nineteen such jumps can be encoded in a 64-bit integer identifier, capable of identifying more than 10**18 distinct locations. As mentioned above, this is more than adequate for any practical purpose, as each such facet occupies only a couple of square inches on the earth's surface. The fact that most computers cannot represent integers of such magnitude may be raised as an objection to using such codings, but there is never a need to manipulate a complete geocode at once, except to copy it. This can be done piecewise.

What About the Geoid?

More serious objections to GEM encoding can be raised concerning the assumptions necessary to model a planet as a polyhedron. Real celestial bodies are not perfect spheres, and their geodetic irregularities introduce disparities between the locations of a regular polyhedron's vertices and regularly-spaced points on the planet's surface. Minor departures from sphericity can noticeably displace the computed locations of vertices at large scales. This would negate much of the spatial utility and universality of GEM geocoding.

Fortunately, geoidal variations can be represented by geodesic models. All that should be necessary is to orient the initial octahedron to cardinal points, and subdivide it a few times into several thousand facets. Their vertices can then be projected onto the geoid by applying the relevant geoid model. The transformed coordinates would then be archived to serve as control for all further subdivisions, in effect serving as a finite-element model of the geoid. This basic framework could then be made available to the public in a standardized format, providing unified ground control for any application in any location. Should geodetic revisions occur, the 'standard' faceting would have to be recomputed, but user datasets would not have to be edited. That is, new assumptions about the shape of the planet would alter the locations represented by geocodes but leave the geocodes themselves unchanged. This somewhat facile description hides the enormous amount of labor that may be needed to create a unified model of the geoid. Intense commitment and cooperation between public agencies and

FIGURE 4. *A six-level GEM breakdown illustrating geocoding properties; note that only octahedral facets converging on Gemcode 69825846 are shown.*

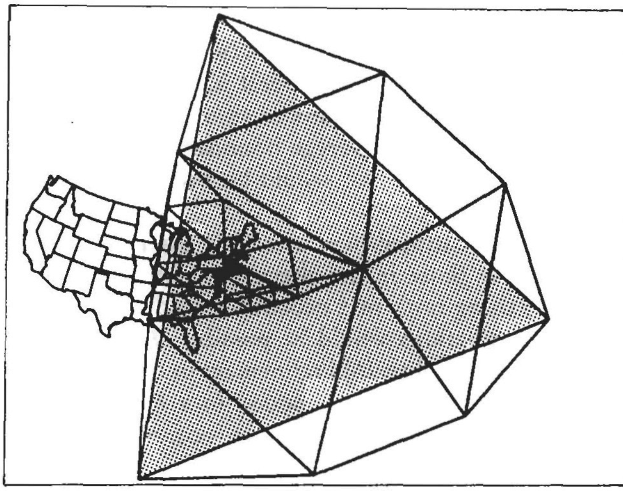


FIGURE 4a. *United States in relation to* GEM *octant 6......*

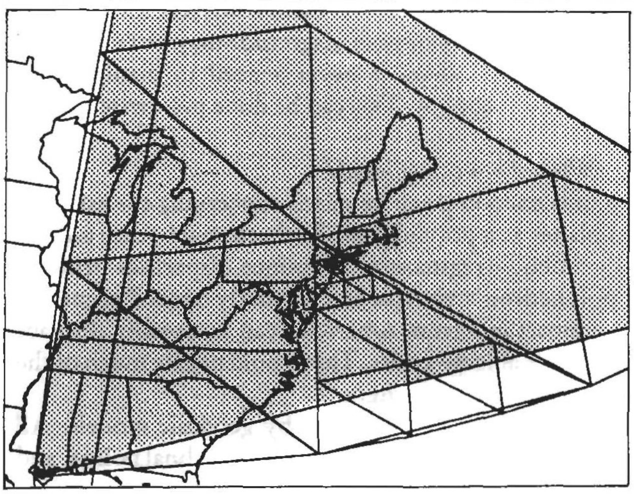


FIGURE 4b. *Eastern United States within* GEMCODE *69......*

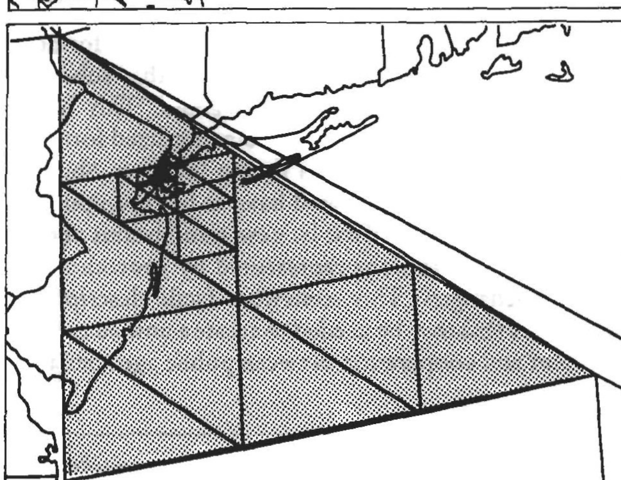


FIGURE 4c. *Northeastern United States centered on* GEMCODE *6982....*

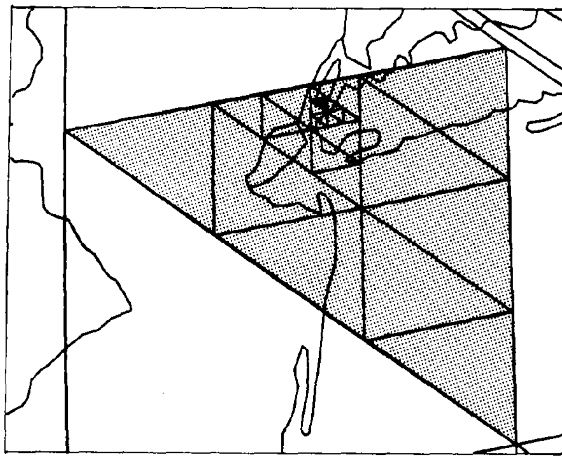


FIGURE 4d. *New York and New Jersey centered on* GEMCODE *69825...*

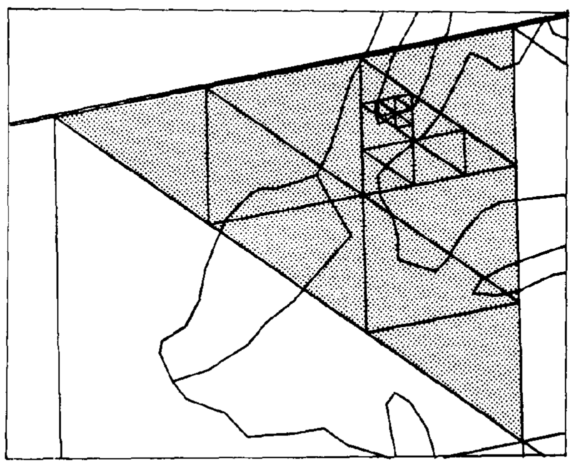


FIGURE 4c. *Part of Greater New York:* GEMCODE *698258..*

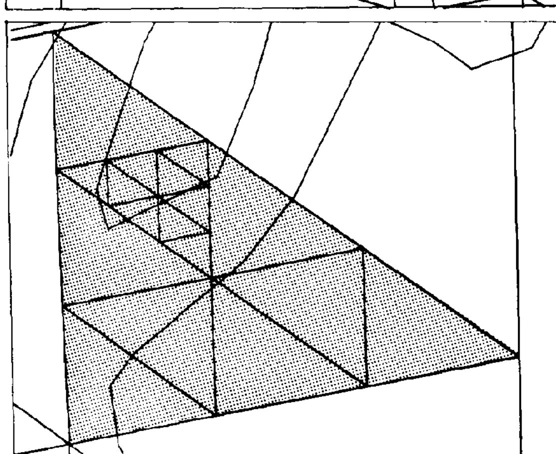


FIGURE 4f. *Lower Manhattan and Brooklyn:* GEMCODE 6982584.

professional societies in many nations appears to be needed to do the job right. Using a geodesic approach the actors can negotiate global differences without obsoleting entire map series or databases, which rather than having to be recompiled, can be periodically tuned into better shape.

FORWARD

Viewed from space, the Earth assumes the form of a beautiful, variegated bubble. In our efforts to measure and model its topography, we forget that as rugged as landforms may appear to us here, they are merely its texture. Were we able to compress it to be one meter across, the Earth would feel as smooth as the proverbial Billiard Ball: Mount Everest would protrude just half a millimeter above the spheroid. GEM offers a way to model the form of our planet or others, gracefully and economically, and as completely as one is prepared to measure and encode its surface.

In the course of several years, GEM has evolved[5] from a space-saving scheme to encode gridded surfaces as quadtrees (its DEPTH component) into a fractal formulation of global geomorphology. Throughout the process, useful properties continued to emerge while the model's size shrank and its simplicity grew. Work remains to be done, especially in regard to the applications (archival, analytic, graphic) that GEM might effectuate or enhance. As well as circumstances have allowed, the origins, procedures and properties of the model have been discussed. Although certain suggestions were made, no particular organization for data was prescribed. This is due both to unanswered questions about the model and to the inevitable tradeoffs involving memory capacity, file access and other properties of particular systems.

While any given implementation of GEM is likely to be concerned with local terrain data, any of these archives can be merged into larger databases. Their union can be made to contain the most accurate versions of any regions which happen to be duplicated. It is therefore not unrealistic to envision a network of GEM archives, each with detailed data concerning certain localities, functioning as a coherent, reliable but decentralized database.

Claiming that a data model as minimal as GEM can catalog and verify vast numbers of measurements more or less automatically is likely to raise some eyebrows. It is hoped that this presentation has communicated how such properties derive by construction and inference from the polyhedral and polynomial structures serving as the basis for GEM. All persons wishing to comment, contest or collaborate, please contact the author. Together, we may be able to express ideas and information as images of our planet, illuminating its form, features and facets, in all their fullness.

NOTES

[1] An icosahedron is a polyhedron having 20 equilateral triangular faces joined along 30 edges which connect each of its 12 vertices to 5 neighboring ones. It is the largest regular platonic solid which is convex (not stellated). This figure is the basis for the construction of a large family of space-frame structures developed by R. Buckminster Fuller, his colleages and followers. For a comprehensive statement of the principles governing these constructions consult Fuller, 1982.

[2] The duality of the cube and octahedron is central to Fuller's geometrics. One of his most interesting discoveries is a construction known as the 'jitterbug', which demonstrates the instability of the cube-oct form, which Fuller terms 'vector equilibrium'. The jitterbug swings ambivalently from an octahedral state through a cube-octahedron stage before collapsing again into a re-aligned octahedron, pumping in and out in alternating cycles. GEM's geometry comes quite close to being a high-frequency jitterbug.

[3] These first eight levels of detail do not contain much information about the relief of the Earth, but do reflect its shape. If not encoded, this hierarchy assumes spherical proportions. If encoded with geodetic data, however, the 52,488 facets can model the geoidal shape quite adequately. Construction of such a model geoid alone might make the investment in developing GEM worthwhile. This possibility is discussed below.

[4] This, of course, requires that coordinates be expressed as, or be converted to, geodetic coordinates; local coordinate systems are not directly translatable into GEM control points.

[5] A number of individuals have contributed to this process. Without question, much of its energy devolves from Buckminster Fuller's work, amplified by innumerable other ideas. Among those are Benoit Mandelbrot's elegant expression of elemental eclecticism, *les formes fractals, ce n'est pas*; while barely hinted at here, this perspective provides important criteria for understanding what the model can contain. Closer to home, Denis White has patiently volleyed ideas off walls where I lobbed them. Dennis Dreher's geometric competence has helped to keep the scheme rooted in physical reality, in the process of constructing a scale model of the data structure. Kelly Chan is gratefully acknowledged for his musings, essays and especially, code. To Dan Schodek, Faculty Director of the Lab for Computer Graphics, goes my appreciation for his tolerance and support of visionary puttering. The author, notwithstanding, assumes full responsibility for GEM, in all its polymorphic perversity.

REFERENCES

DUTTON, G.H. 1983. Efficient encoding of gridded surfaces. *Spatial algorithms for processing land data with a microcomputer.* Cambridge, Ma: Lincoln Institute for Land Policy Monograph Series.

FULLER, R.B., with E.J. APPLEWHITE. 1982. *Synergetics: explorations in the geometry of thinking.* New York: MacMillan, 876 p.

FULLER, R.B. and S. SADAO. 1982. *Spaceship Earth edition of the Dymaxion sky-ocean map.* Philadelphia: Buckminster Fuller Institute.

FOWLER, R.J. and J.J. LITTLE. 1979. Automatic extraction of irregular network digital terrain models. *Computer Graphics*, vol. 13, no. 2, pp. 199–207.

GARGANTINI, I. 1982. An effective way to represent quadtrees. *Comm. of the ACM.* vol. 25, no. 12, pp. 905–910.

GOLD, C.M. 1978. The practical generation and use of geographic triangular element data structures. *Harvard Papers on Geographic Information Systems.* Reading, Ma: Addison-Wesley, vol. 5.

MALES, R.M. 1978. ADAPT: A spatial data structure for use with planning and design models. *Harvard Papers on Geographic Information Systems.* Reading, Ma: Addison-Wesley, vol. 5.

MANDELBROT, B. 1982. *The fractal geometry of Nature.* San Francisco: Freeman, 462 p.

MARK, D.M. 1978. Concepts of data structure for digital terrain models. *Proc. of the Digital Terrain Model Sy sium.* Falls Church, Va: Amer. Soc. Photogrammetry, pp. 24–31.

PEUCKER, T.K., R.J. FOWLER, J.J. LITTLE and D.M. MARK. 1977. *Digital representation of three-dimensional surfaces by triangulated irregular networks.* Tech. Rep. 10, ONR Contract no. N00014-75-C-0886, Dept. of Geography, Simon Fraser U., Burnaby, B.C., Canada.

POPKO, E.S. 1968. *Geodesics.* Detroit: University of Detroit Press.

PREY, A. (B. Binder, trans., 1968). A mathematical representation of the altitude relationships on the surfaces of the earth, using spherical functions of the 16th order. *Harvard Papers in Theoretical Geography.* No. 22. Originally published in German, 1922.