(to appear in: (1998) *Computing Science and Statistics*, 30, ed. S. Weisberg, Interface Foundation of North America, Inc., Farfax Station, VA.)

Discrete Global Grid Systems*

Kevin Sahr[1] and Denis White[2]

[1]Department of Computer and Information Science, University of Oregon, Eugene, Oregon, USA
[2]US EPA NHEERL Western Ecology Division, Corvallis, Oregon, USA

Abstract

A new class of spatial data structures called discrete global grid systems (DGGS's) is introduced and the general application classes for it are discussed. DGGS's based on subdivisions of the platonic solids, called Geodesic DGGS's, are then introduced. A number of existing and proposed Geodesic DGGS's are examined by looking at four design choices that must be made in constructing a Geodesic DGGS: the base platonic solid, the orientation of that solid relative to the earth's surface, the method of subdivision defined on a face of that solid, and a method for relating that planar subdivision to the corresponding spherical surface. Finally, an examination of these design choices leads us to the construction of the ISEA3H DGGS.

Key Words: discrete global grid systems, spatial data structures.

Introduction

The use of spatial data structures based upon regular hierarchical partitions of subsets of the plane has become increasingly common. Such data structures exhibit a high degree of regularity that makes it possible to develop very efficient algorithms for common spatial database and geometric operations (Nievergelt, 1989; Samet, 1990).

One of the more important spatial domains is the surface of the earth. This domain is of course topologically equivalent to the surface of a sphere and not to a subset of the cartesian plane. However, traditionally the earth's surface has been transformed into a subset of the cartesian plane using map projections, and spatial data structures have been built referenced to these planar spaces. This approach has proven satisfactory for many applications, particularly those that deal with only a portion of the earth's surface. But as truly global data sets of high resolution have become increasingly available, along with the computing power necessary to manipulate them on a global scale, there have been increasing efforts to develop data structures which more closely retain the topology of the earth's surface. In this paper we will discuss some of the more promising attempts to extend the concept of data structures based on regular hierarchical partitions to global data sets. We will begin by defining the class of spatial data structures under consideration, which we call *discrete global grid systems* (DGGS's), and discussing the major classes of applications for which they have been proposed. We will then look at some of the design choices which have been made in developing various existing and proposed DGGS's.

Discrete Global Grid Systems

Let a *discrete grid* be a set of areal cells that form a partition of the spatial domain of interest, with each cell having a point associated with it. Depending on the application, vectors of data values may be assigned to either the cells, the points, or to the point/cell combinations. Let a *discrete grid system* be a (possibly infinite) series of discrete grids on the same spatial domain. A *discrete global grid system* (DGGS) is a discrete grid system in which the domain of interest is the earth's surface, usually represented by some form of topologically equivalent approximation such as a sphere, a spheroid, or a geoid.

Usually the series of discrete grids which constitute the discrete grid system are a series of increasing resolution grids. If the resolution increases in a regular fashion we can define the *aperture* of the system as the number of resolution k+1 cells that correspond to a single resolution k cell, where the resolution k+1 discrete grid is one resolution finer than the resolution k grid. (This definition is a generalization of the one given in Bell *et al.*, 1983). Later we shall discuss some DGGS's which have more than one type of cell. In these cases there is always one cell type which clearly predomi-

*The information in this article has been funded in part by the United States Environmental Protection Agency, under cooperative agreement CR821672 with Oregon State University. It has been subjected to Agency review and approved for publication. The conclusions and opinions are solely those of the authors and are not necessarily the views of the Agency. Additional funding has been provided by the Biodiversity Research Consortium under cooperative agreement PNW-92-0283 with the USDA Forest Service.

nates, and the aperture of the system is defined using the dominant cell type.

The types of data objects that are stored in spatial data structures are often considered as falling into two classes (e.g., Chrisman, 1997). These have been referred to — though sometimes with subtle distinctions — under a variety of terms in different contexts:

fields	entities/objects
raster	vector
space filler	space bounding
regions	points/locations
interiors	borders

The first class of objects, as represented in the first column, are those where vectors of data values are associated with 2-dimensional areal cells. Often the data of interest is some function defined over the spatial domain, and each cell represents a region over which the function takes on a uniform or nearly uniform value, that value being the primary data attribute of the cell.

Representations in the second column are classified as dimensionless point or location data and objects with distinct boundaries, in particular ones that can be defined as a series of 1-dimensional line segments referenced in terms of their end-points. Under this representation vectors of data values are associated with points, or with entities defined in terms of points.

Resolution and Graticule

The familiar latitude/longitude graticule is the most common basis for DGGS's in use today. In the case of DGGS's for storing field data, lines spaced at regular latitude and longitude increments form the boundaries of areal cells. Figure 1 shows such a grid with a 10^0 spacing between lines of latitude and longitude.

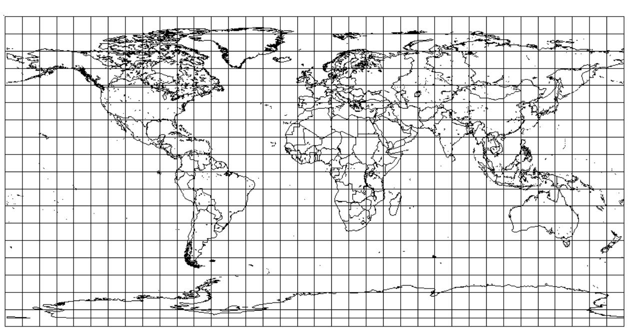

Figure 1. Field discrete grid induced by latitude/longitude graticule with 10^0 resolution.

A DGGS may be formed by creating additional resolution discrete grids by grouping (or subdividing) squares to form squares of coarser (or finer) resolution. A square can be divided into n^2 smaller squares by breaking each edge of the original square into n pieces and connecting the break points with lines parallel to the original edges. The most common such structure is the quadtree (where n = 2), which has seen use in a variety of computer science applications (as summarized in Samet, 1990). In a quadtree each square at a given resolution is subdivided into four squares to form the next finer resolution grid. The quadtree is a specific instance of an r^d-tree (Nievergelt, 1989) with dimension d = 2 and radix r = 2. That is, it has a two-dimensional domain, and it is divided by a factor of two in each dimension at each level in the tree. r^d-trees have also been applied to grids based on the latitude/longitude graticule. For example, we might define an r^d-tree with d = 2 and r = 10 to form successive grid resolutions corresponding to the digits of a decimal degree representation. Figure 2 shows a portion of the next resolution (1^0) of such a structure formed on the grid depicted in Figure 1. Each of the 10^0 cells divides evenly into 100 1^0 cells (10 in each of the two dimensions of latitude and longitude), and this subdivision can be continued to an arbitrarily fine resolution, corresponding to an arbitrary precision in decimal degrees. This system has an aperture of 100.

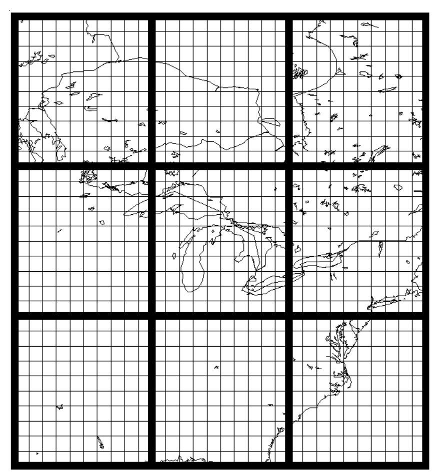

Figure 2. Portion of two resolutions (10^0 and 1^0) of a field r^d-tree (d = 2, r = 10) defined by the latitude/longitude graticule.

While not illustrated above, field grids usually associate with each of the cells a point location that can be used to facilitate spatial access to, and spatial operations on, the areal cells. For example, the distance between two cells might be defined as the distance between their associated points. Usually these points are placed consistently at the center or at an essentially arbitrary vertex of the cell.

Discrete grid systems based on r^d-trees have a number of useful properties. For example, one of the most fundamental problems in spatial database theory is that the two-dimensional spatial problem domain must be stored in a linear address space in memory or on disk. This means that not all of the spatial locality information in the actual domain can survive the transformation to computer storage. R^d-trees have proven convenient for developing linearizing space-ordering methods that preserve a high degree of spatial locality, such as Morton ordering (a discussion of a number of these methods is given in Goodchild and Grandfield, 1983). Also, r^d-trees can be used to induce hierarchical 1-dimensional addresses upon which can be defined *tesseral arithmetics*. Tesseral arithmetics (see, for example, Diaz, 1984) provide for extremely efficient computation of many geometric operations directly on the addresses themselves. Finally, homogeneous regions of the domain, where the data value of concern remains uniform over a wide region, may be represented by a coarser resolution grid than areas where the value changes more abruptly. Thus the subdivision need not be performed uniformly across the domain, saving on overall storage requirements.

DGGS's have also been used for storing locational data. As pointed out by Dutton (1996), the traditional latitude/longitude locational system can be thought of as a DGGS, with each digit of precision constituting a discrete grid resolution, and the resolution of a location being determined by the number of significant digits (though he also notes that such a device is rarely used in practice; usually only a single resolution cartesian locational grid is used in a given application). In such systems point grids are formed by taking the intersections of regularly spaced lines of latitude and longitude (these lines formed the boundaries of the areal cells in the field grid described above). Locations in the original spatial domain are then mapped to the nearest point. The square voronoi regions associated with each of the points form cells that determine which locational regions are mapped to which points. Such a grid is in effect a field grid where the data value of concern is location; points of "nearly uniform" location are mapped to the same cell in the grid. Figure 3 illustrates a portion of a 10^0 resolution location discrete grid induced by the latitude/longitude graticule.

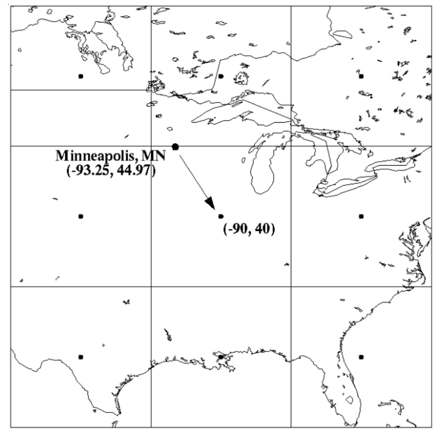

Figure 3. Portion of 10^0 resolution location discrete grid induced by the latitude/longitude graticule. Here the actual location of Minneapolis, MN is assigned a 10^0 resolution grid location as shown.

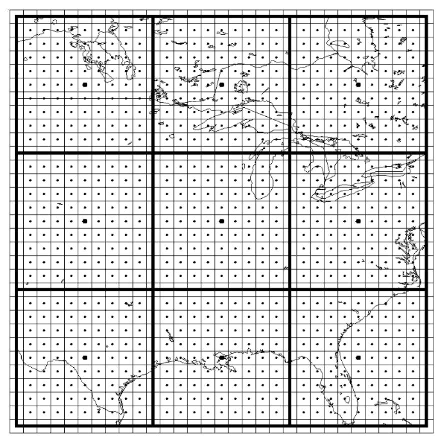

Figure 4. Portion of two resolutions (10^0 and 1^0) of a location DGGS defined by the latitude /longitude graticule.

Changing resolution induces another discrete grid; Figure 4 shows a portion of the 1^0 resolution grid corresponding to the 10^0 resolution grid depicted in Figure 3. It should be noted that this location grid is not the same as the field grid system depicted earlier. The latitude/longitude lines play a different role in each (in the field grid they form the cell boundaries, while in the locational grid their intersections form the point grid). Also, in the field grid the boundaries of the 1^0 cells coincide with the boundaries of the 10^0 grid cells; each of the coarser resolution cells can be thought of as an aggregate of 100 finer resolution cells. This is not the case in the locational grid; in this grid the coarse boundaries are straddled by finer resolution cells.

We formalize this distinction as follows. Let G be an aperture n discrete grid system where each resolution k cell is the union of n resolution k+1 cells. We say that G has the property of being *congruent*. If G does not have this property, then we say that G is *incongruent*. Thus the graticule-based field grid is congruent, as are all r^d-tree structures such as quadtrees. The graticule-based location grid is incongruent. Both data structures are, however, discrete global grid systems with an aperture of 100.

In addition to their use as static data structures for field and location data, two specific applications have been proposed for DGGS's. These systems have been used to develop statistically sound survey sampling designs on the earth's surface. A discussion of this application is given in Olsen *et al.* (1998) and will not be further discussed here. DGGS's have also been proposed as the basis for spatially discrete dynamic simulations such as those used in global climate modeling (Williamson, 1968; Sadournay *et al.*, 1968; Heikes and Randall, 1995a, 1995b; Thuburn, 1997).

DGGS's based on the latitude/longitude graticule have numerous practical advantages. They are based on a 2-dimensional cartesian coordinate system; such systems have long been a foundation of scientific inquiry on spatial domains. Such square-based grids also map easily to common display devices. The latitude/longitude system itself has been used extensively since well before the computer era and is therefore the basis for a wide array of existing data sets, processing algorithms, and software.

But such grids also have limitations. Square grids in general do not exhibit uniform adjacency. Each square grid cell has four neighbors with which it shares an edge, and whose centers are equidistant from it's center. But each cell also has four neighbors with which it shares only a vertex, and whose centers are a different distance from its center than the distance to the centers of the edge neighbors. This is particularly a problem when we attempt to use such grids for discrete simulations.

DGGS's based on the latitude/longitude graticule become increasingly distorted in area and shape as one moves north and south from the equator. The north and south poles, both points on the surface of the globe, map to lines in the latitude/longitude system; the top and bottom row of grid cells in Figure 1 are in fact triangles, and not squares as they appear on the plane. These polar singularities have forced applications such as global climate modeling to make use of special grids for the polar regions. Finally, DGGS's induced by the latitude/longitude graticule do not have equal area cells, which is important for many applications.

Geodesic DGGS's

The inadequacies of DGGS's based on the latitude/longitude graticule has led a number of researchers to explore alternative approaches. As has been widely observed (e.g., White *et al.*, 1992) the spherical versions of the five platonic solids represent the only ways in which the sphere can be partitioned into cells each consisting of the same regular spherical polygon, with the same number of polygons meeting at each vertex. A number of researchers have used the platonic solids as a starting point for further recursive subdivisions to build finer resolution discrete grids, and in our opinion these attempts have led to the most promising known options for DGGS's. A number of these researchers have been inspired directly or indirectly by R. Buckminster Fuller's work in descretizing the sphere that led to his development of the geodesic dome, and for this reason we will refer to this class of DGGS's as *Geodesic DGGS's*.

It would be premature to conclude that any one proposed Geodesic DGGS is the ideal DGGS. Indeed, it may well be the case that no single DGGS will ever prove optimal for all applications. Many of the proposed systems include design innovations in particular areas, though their construction may have involved other, less desirable design choices. Therefore, rather than surveying individual Geodesic DGGS's as monolithic, closed systems, we will take the approach here of viewing the construction of a Geodesic DGGS as a series of design choices which are, for the most part, independent.

The following design choices must be made to fully specify a Geodesic DGGS:

1. The base platonic solid.
2. The orientation of the base platonic solid relative to the earth.
3. The hierarchical spatial partitioning method defined symmetrically on a face (or set of faces) of the base platonic solid.
4. The transformation between each face and the corresponding spherical surface.

We will now look at each of these design choices in turn, discussing the choices made in the development of a number of Geodesic DGGS's.

Choice of Base Platonic Solid

The five platonic solids are shown in Figure 5. In general, the greater the number of faces in the base platonic solid chosen, the less the distortion introduced when projecting between a face of the polyhedron and the corresponding spherical surface. The icosahedron has the greatest number of faces (20) and therefore projections defined on it tend to have relatively small distortion. Thus the icosahedron is the most common choice for base platonic solid. Geodesic DGGS's based on the icosahedron include those of Williamson (1968), Sadournay *et al.* (1968), Fekete and Treinish (1990), Thuburn (1997), and (with a slight adjustment as discussed below) Heikes and Randall (1995a, 1995b).

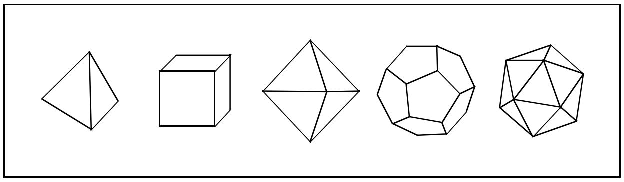

Figure 5. The platonic solids: the tetrahedron, cube, octahedron, dodecahedron, and icosahedron.

The tetrahedron and cube have the smallest number of faces (4 and 6 respectively) and are thus relatively poor base approximations for the sphere.

The octahedron was chosen as the base platonic solid for what is arguably the most currently well-developed Geodesic DGGS, the Quaternary Triangular Mesh (QTM) system of Dutton (1988, 1996). The octahedron has the advantage that it can be oriented with vertices at the north and south poles, and at the intersection of the prime meridian and the equator. In this orientation the faces of the octahedron align with the spherical octants formed by the equator and the prime meridian. Given a point in latitude/longitude coordinates it is thus trivial to determine which octahedron face the point lies on. As noted above, though, because it has fewer faces than the icosahedron projections defined on the faces of the octahedron tend to have higher distortion (White *et al.*, in press).

Finally, Wickman *et al.* (1974) observe that if a point is placed in the center of each of the faces of a dodecahedron and then raised perpendicularly out to the surface of the sphere ("stellated"), each of the 12 pentagonal faces becomes 5 isosceles triangles. The stellated dodecahedron thus has 60 triangular faces compared to the 20 faces of the icosahedron, and thus a projection can be defined on the stellated dodecahedron with lower distortion than on the icosahedron (e.g., Snyder, 1992). However, there is a trade-off in that the base triangles are no longer equilateral and the act of stellation complicates the base polyhedral structure.

Choice of Polyhedron Orientation

Once a base platonic solid is chosen, an orientation relative to the actual surface of the earth must be specified. One compact way of specifying this is by giving the geodetic coordinates of one of the polyhedron's vertices and the azimuth from that vertex to an adjacent vertex. For regular platonic solids this information will completely specify the position of all the other vertices.

As mentioned above, Dutton (1988, 1996) orients the octahedron so that its faces align with the octants formed by the equator and prime meridian.

Wickman *et al.* (1974) orient the dodecahedron by placing the center of a face at the north pole and a vertex of that face on the prime meridian, thus aligning with the prime meridian an edge of one of the triangles created by stellating the dodecahedron.

In the case of the icosahedron, the most common orientation is to place a vertex at each of the poles and then align one of the edges emanating from the vertex at the north pole with the prime meridian. This orientation is used in the systems of Williamson (1968), Sadournay *et al.* (1968), Fekete and Treinish (1990), and Thuburn (1997). Figure 6a illustrates this orientation.

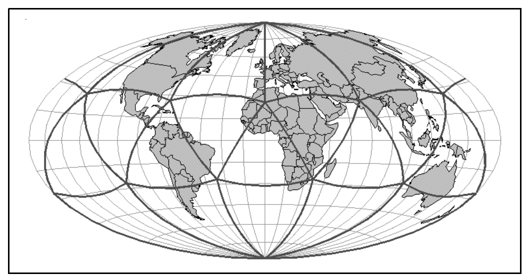

Figure 6a. Spherical icosahedron orientation with vertices at poles and edge aligned with prime meridian.

Fuller (1975) developed an icosahedron orientation for his Dymaxion icosahedral map projection. He placed all 12 of the icosahedron vertices in the ocean so that the icosahedron could be unfolded onto the plane without breaks in any land mass. This is the only known placement with this property. Figure 6b shows Fuller's orientation.

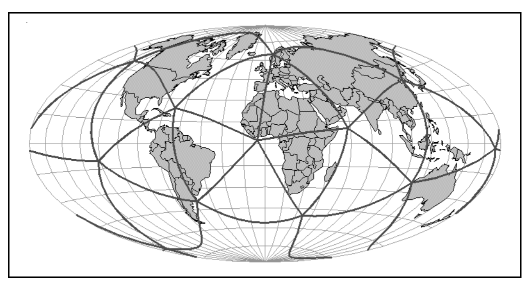

Figure 6b. Fuller's spherical icosahedron orientation with all vertices in oceans.

The icosahedron-based system of Heikes and Randall (1995a, 1995b) was developed specifically for performing fluid flow simulations of global climate. They noted that in the most common icosahedron placement (figure 6a) the icosahedron is not symmetrical about the equator. When a simulation on a DGGS with this orientation is initialized to a state symmetrical about the equator, and then allowed to run, it evolves into a state which is asymmetrical about the equator, presumably due to the asymmetry in the underlying icosahedron orientation. To counter this they rotated the southern hemisphere of the icosahedron 36^0, and the resulting "twisted icosahedron" is symmetrical about the equator.

We have noted that if the icosahedron is oriented so that the north and south poles lie on the midpoints of edges rather than at vertices then it is symmetrical about the equator without further adjustment. Since under some subdivision schemes there are special-case cells at the vertices (see below), following Fuller's lead we attempted to minimize the number of vertices which fall on land, so that at least land-based applications might be able to avoid the special case handling of vertex cells. We discovered an orientation which has only one vertex on land, in China's Sichuan Province. This orientation is shown in Figure 6c.

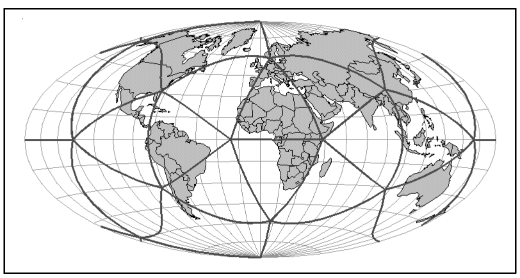

Figure 6c. Spherical icosahedron oriented for symmetry about equator by placing poles at edge midpoints.

Choice of Spatial Partitioning Method

Beginning with the chosen base platonic solid, we must now choose a method of subdividing this polyhedron to create finer resolution discrete grids. It is only necessary to define the subdivision methodology on a single face of the polyhedron, or some set of faces which constitute a unit which tiles the polyhedron, provided that the subdivision is symmetrical with respect to the face or tiling unit.

We have seen that the preferred choices of base platonic solid are the icosahedron, the octahedron, and the stellated dodecahedron, each of which has a triangular face. Beginning with a triangle, the obvious choice for further subdivision is to divide the triangle into smaller triangles. Like the square, an equilateral triangle can be divided into n^2 smaller equilateral triangles by breaking each edge into n pieces and connecting the break points with lines parallel to the triangle edges. In the literature of geodesic domes this is referred to as a *Class I* or *alternate* breakdown (Kenner, 1976). Recursively subdividing the triangles thus obtained in the same manner yields an aperture n^2 discrete grid system. Small apertures have the advantage of allowing more potential grid resolutions so that applications can choose a resolution which best meets their needs. In the case of triangle subdivision the smallest possible aperture is 4 (n = 2). This aperture is also convenient because it parallels the breakdown of the square grid quadtree, and many of the algorithms developed on the square grid quadtree are transferable to the triangle grid quadtree with only minor modifications (Fekete and Treinish, 1990). Figure 7 shows this subdivision approach, which is used in the DGGS's of Wickman *et al.* (1974), Dutton (1988, 1996), and Fekete and Treinish (1990).

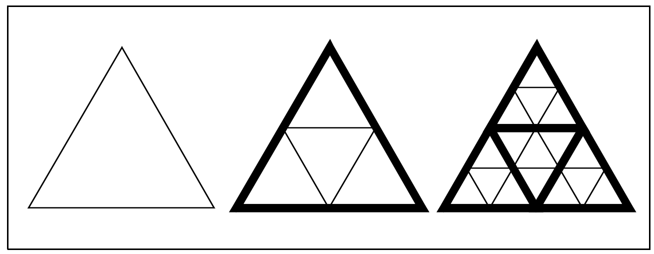

Figure 7. Three levels of a Class I/Alternate aperture 4 triangle subdivision.

It should be noted that another aperture 4 triangle subdivision is possible. In this approach, referred to as the *Class II* or *triacon* breakdown (Kenner, 1976), each triangle edge is broken into $n = 2^m$ pieces (where m is some positive integer). Lines are then drawn perpendicular to the triangle edges to form the new triangle grid. This breakdown is illustrated in Figure 8. Note that while the Class I/alternate breakdown is congruent, the Class II/triacon breakdown is incongruent.

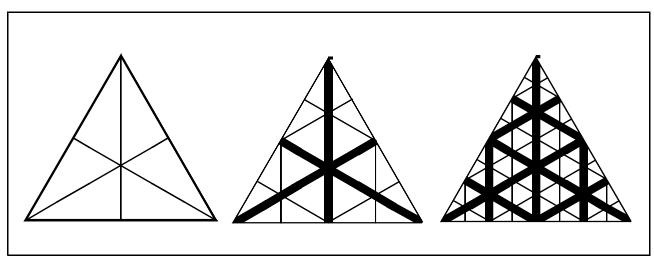

Figure 8. Three levels of a Class II/Triacon aperture 4 triangle subdivision.

Triangles have a number of disadvantages as the basis for a discrete grid system. First, they are not squares; they are thus a foreign alternative for many potential users, and they do not display well on common output display devices that are based on square lattices of pixels. Like square grids, they do not display uniform adjacency, each cell having three edge and nine vertex neighbors. Unlike squares, the cells of triangle-based discrete grids do not have uniform orientation; as can be seen in Figure 7, some triangles point up while others point down. Many algorithms defined on triangle grids must therefore keep track of triangle orientation.

The hexagon has received a great deal of recent interest as a potential basis for discrete grid systems. Hexagons are the most compact regular polygon which can tile the plane, and hexagonal grids provide the greatest angular resolution when compared to square and triangular grids (Golay, 1969). Unlike square and triangle grids, hexagon grids do have uniform adjacency; each hexagon cell has six neighbors, all of whose centers are exactly the same distance away from its center. This fact alone has made them increasingly popular as bases for discrete spatial simulations. The case for their superiority for this purpose was greatly strengthened when Frisch *et al.* (1986) discovered that a discrete hexagonal lattice-gas model of incompressible hydrodynamics asymptotically goes over to the continuous two-dimensional Navier-Stokes equations of incompressible hydrodynamics. This is a surprising and not at all intuitive result; the six discrete velocity vectors of the hexagonal lattice are necessary and sufficient to simulate continuous, isotropic, fluid flow. A recent textbook (Rothman and Zaleski, 1997) on fluid flow cellular automata is based entirely on hexagonal meshes, with discussions of square meshes included “only for pedagogical calculations.” Triangle grids, which are even more insufficient for this purpose, are not mentioned.

While single resolution hexagon-based discrete grids are becoming increasingly popular, the use of multi-resolution hexagon-based discrete grid systems has been hampered by the fact that congruent discrete grid systems cannot be built using hexagons; it is impossible to exactly decompose a hexagon into smaller hexagons (or, conversely, to aggregate small hexagons to form a larger one). Hexagons can be aggregated in groups of seven to form coarser resolution objects which are almost hexagons, as illustrated in Figure 9, and these can again be aggregated into pseudo-hexagons of even coarser resolution, and so forth. This structure has a very efficient tesseral arithmetic defined on it called *generalized balanced ternary* (Gibson and Lucas, 1982), and because of this it has become the most widely used multi-resolution hexagon-based grid system. However, it has several problems as a general purpose basis for spatial data structures. The first is that the cells are hexagons only at the finest resolution. Secondly, the finest resolution grid must be determined prior to creating the system, and once determined it is impossible to extend the system to finer resolution grids. Thirdly, the orientation of the tessellation rotates by about 19 degrees at each level of resolution. Finally, it does not appear to be possible to symmetrically tile a triangular face with such a hierarchy, which makes it unusable as a subdivision choice for a Geodesic DGGS.

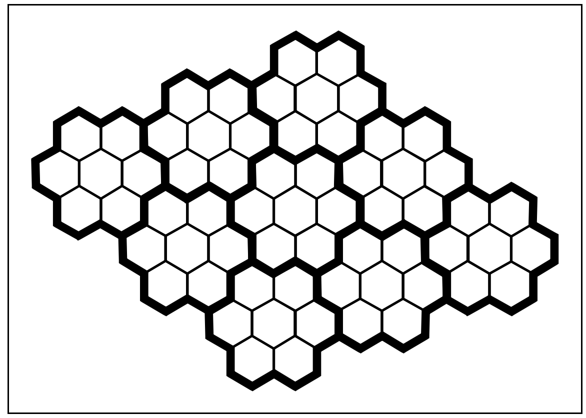

Figure 9. Seven-fold hexagon aggregation into coarser pseudo-hexagons.

There are however an infinite series of apertures that produce regular hierarchies of incongruent hexagon discrete grids. Figure 10 shows three levels of an aperture 3 hexagon subdivision, the smallest such subdivision possible.

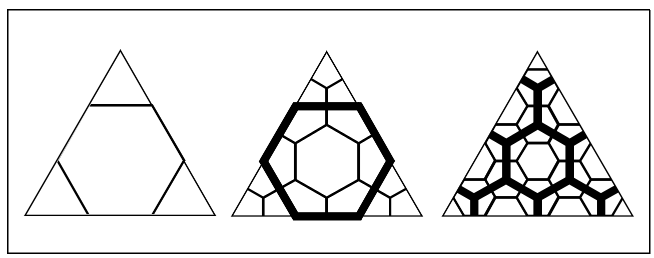

Figure 10. Three levels of an aperture 3 hexagon subdivision.

Aperture 4 hexagon subdivisions are also possible. Figures 11 and 12 illustrate aperture 4 hexagon subdivisions corresponding to the Class I/alternate and Class II/triacon symmetry axes respectively. The DGGS’s of Heikes and Randall (1995a) and Thuburn (1997) are aperture 4 Class I/alternate hexagon grids, while Williamson (1968) uses an aperture 4 Class II/triacon hexagon grid. Sadournay *et al.* (1968) use a Class I/alternate hexagon grid of arbitrary aperture.

It should be noted that it is impossible to completely tile a sphere with hexagons. When a base polyhedron is tiled with hexagon-subdivided triangle faces a non-hexagon polygon will be formed at each of the polyhedron’s vertices. The number of such polygons, corresponding to the number of polyhedron vertices, will remain constant regardless of grid resolution. In the case of an octahedron these polygons will be 8 squares, in the case of the icosahedron they will be 12 pentagons.

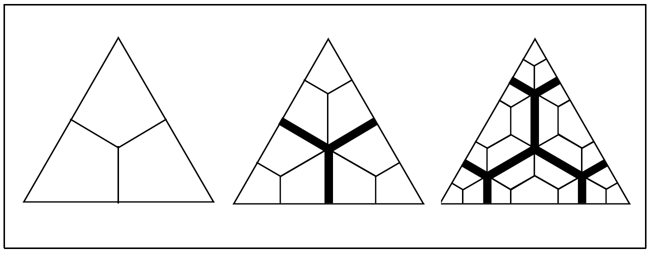

Figure 11. Three levels of an aperture 4 Class I/Alternate hexagon subdivision.

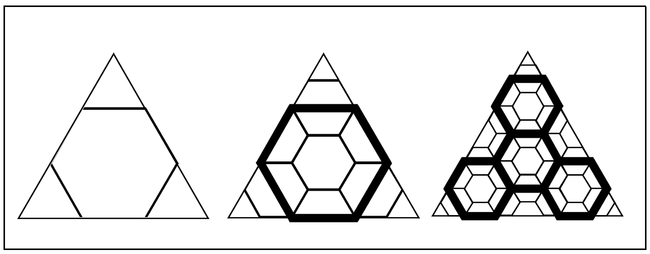

Figure 12. Three levels of an aperture 4 Class II/Triacon hexagon subdivision.

Choice of Transformation

Once a method has been chosen for subdividing the planar face of the polyhedron some transformation must be chosen for creating a similar topology on the corresponding spherical surface. Perhaps the simplest approach is to perform the subdivision directly on the spherical surface, using great circle arcs corresponding to the lines in the plane. The aperture 4 Class I/alternate triangle subdivision can be performed on the sphere by connecting the midpoints of the edges of the base spherical triangle, and then recursively performing the same operation on each of the resulting triangles. This technique is used by Fekete and Treinish (1990).

It is important to note, however, that the three sets of great circle arcs corresponding to the grid lines of the planar triangle subdivision methods shown above do not in general intersect in points on the surface of the sphere as they do on the plane. Numerous methods for constructing triangle subdivisions on the sphere analogous to the Class I and Class II plane subdivisions are given in Clinton (1971) and Kenner (1976).

Williamson (1968) uses great circle arcs corresponding to two of the three sets of Class II/triacon triangle subdivision grid lines to determine a set of triangle vertices, and then forms the last set of grid lines by connecting the existing vertices with great circle arcs. These triangle vertices form a dual aperture 4 Class II/triacon hexagon grid.

Sadournay *et al.* (1968) first create an aperture n (where n = m^2 for some integer m) Class I/alternate triangle subdivision. This is constructed on the sphere by breaking each edge of the base spherical triangle into m segments, and connecting the breakpoints of two of the edges with great circle arcs. These arcs are then subdivided evenly into segments corresponding to the planar subdivision. The resulting breakpoints form the centers of a Class I/alternate hexagon grid.

Thuburn (1997) performs an aperture 4 Class I/alternate triangle subdivision and then calculates the spherical voronoi cells of the triangle vertices to form an aperture 4 Class I/alternate hexagon grid.

A number of researchers have attempted to adjust the grids created using great circle arcs to meet some application-specific criteria. For instance, for many applications it would be desirable for the areal cells of each discrete grid resolution to be equal in area; the grids discussed above do not have this property. Wickman *et al.* (1974) begin by connecting the midpoints of the base spherical triangle to form the first resolution of an aperture 4 Class I/alternate grid. They then break each of the new edges at the midpoint into two great circle arcs, and adjust the position of the breakpoint to achieve equal area quasi-triangles. This procedure is then applied recursively to yield an equal-area DGGS. Rather than using great circle arcs for triangle subdivision, Song (1997) proposes using small circle arcs chosen to achieve equal cell areas.

The QTM DGGS (Dutton, 1988, 1996) creates an aperture 4 Class I/alternate triangle subdivision by using small circles corresponding to parallels of latitude relative to the octahedron vertices.

Heikes and Randall (1995a) begin by constructing an aperture 4 Class I/alternate hexagon grid by taking the spherical voronoi of the vertices of an aperture 4 Class I/alternate triangle subdivision on their twisted icosahedron. They then adjust the grid using an optimization scheme to improve its finite difference properties (Heikes and Randall, 1995b) for use in global climate modeling.

White *et al.* (in press) evaluate a number of methods for constructing triangle subdivisions on spherical triangles and observe that using appropriate inverse map projections to transform a subdivided planar triangle into a spherical triangle may be more efficient than using recursively-defined procedures. Projections may be used provided that they map the straight-line triangle edges on the plane to the great-circle arc edges of the corresponding spherical triangle. There are at least three projections with this property. The common gnomonic projection has this property for all polyhedra but exhibits relatively large area and shape distortion. The implementation of Fuller's Dymaxion map projection (Fuller, 1975) given in Gray (1995) also has the required property on the icosahedron but with less area and shape distortion than the gnomonic. Finally, Snyder (1992) gives equal area projections defined on all of the platonic solids, though with greater shape distortion than the Fuller/Gray projection.

An Example: Constructing the ISEA Grid

Now as an example let us take each of the design decisions in turn and attempt to construct a good all-purpose DGGS. First, due to its lower distortion characteristics we choose the icosahedron for our base platonic solid. We orient it with the north and south poles lying on edge midpoints so that the resulting DGGS will be symmetrical about the equator. Next, we note the numerous advantages of hexagon grids discussed above, especially for spatially discrete simulations. While their incongruence may be a disadvantage for storing fields, it is a common feature of multi-resolution location DGGS's such as those induced by the latitude/longitude graticule. We choose the smallest possible hexagon aperture of three. Finally, because equal-area cells are advantageous for many applications, we choose the inverse Snyder equal area projection on the icosahedron to transform the hexagon grid to the sphere. We call the resulting grid the Icosahedral Snyder Equal Area aperture 3 Hexagonal grid (ISEA3H).

Figure 13 shows ETOPO5 5' elevation data binned into four resolutions of the ISEA3H grid by assigning to each cell an elevation value calculated by taking the arithmetic mean of all ETOPO5 data points which fall into that cell.

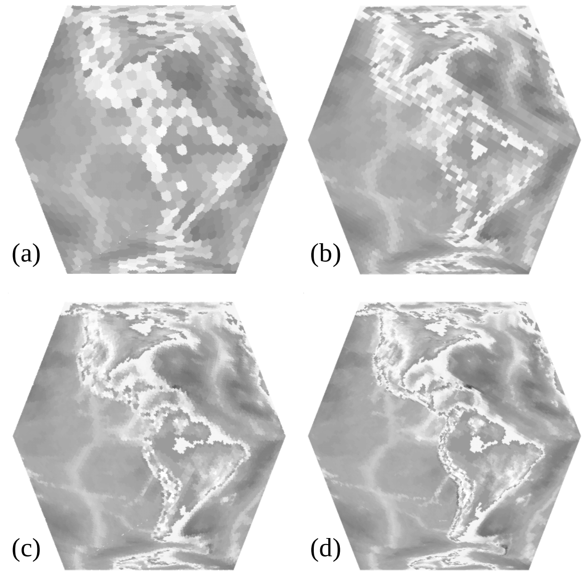

Figure 13. ETOPO5 5' elevation data binned into four resolutions of the ISEA3H DGGS: (a) 210,000 km^2 hexagons, (b) 70,000 km^2 hexagons, (c) 23,000 km^2 hexagons, and (d) 7,800 km^2 hexagons.

References

Bell, S.B., B.M. Diaz, F. Holroyd, and M.J. Jackson (1983). "Spatially referenced methods of processing raster and vector data." *Image and Vision Computing* 1(4): 211-220.

Chrisman, N.R. (1997). *Exploring Geographic Information Systems*. New York: J. Wiley & Sons.

Clinton, J.D. (1971). *Advanced Structural Geometry Studies Part I - Polyhedral Subdivision Concepts for Structural Applications*, Washington, D.C.: NASA CR-1734.

Diaz, B.M. (1984). "Tesseral addressing and arithmetic - overview and theory." Spatial Data Processing Using Tesseral Methods (Diaz, B., and S. Bell, eds.). Reading, UK: Natural Environment Research Council, pp. 1-10.

Dutton, G. (1988). "Geodesic modeling of planetary relief." *Cartographica* 21: 188-207.

Dutton, G. (1996). "Improving locational specificity of map data — a multi-resolution, metadata-driven approach and notation." *International Journal of Geographical Information Systems* 10(3): 253-268.

Fekete, G., and L. Treinish (1990). "Sphere quadtrees: A new data structure to support the visualization of spherically distributed data." *SPIE, Extracting Meaning from Complex Data: Processing, Display, Interaction* 1259: 242-25.

Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). "Lattice-gas automata for the Navier-Stokes equations." *Physics Review Letters* 56: 1505-1508.

Fuller, R. B. (1975). *Synergetics*. New York: MacMillan.

Gibson, L. and D. Lucas (1982). "Spatial data processing using generalized balanced ternary." *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*

Golay, J.E. (1969). "Hexagonal parallel pattern transformations." IEEE Transactions on Computers C-18:8, pp. 733-739.

Goodchild, M.F., and A.W. Grandfield (1983). "Optimizing raster storage: an examination of four alternatives." *Proceedings of Auto-Carto 6*, vol. 1, pp. 400-407.

Gray, R. W. (1995). "Exact transformation equations for Fuller's world map." *Cartography and Geographic Information Systems* 32: 243-246.

Heikes, R. and D. A. Randall (1995a). "Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part I: Basic design and results of tests." *Monthly Weather Review* 123(6): 1862-1880

Heikes, R. and D. A. Randall (1995b). "Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part II: A detailed description of the grid and an analysis of numerical accuracy." *Monthly Weather Review* 123(6): 1881-1887

Kenner, H. (1976). *Geodesic Math and How to Use It*, Berkeley: University of California Press.

Nievergelt, J. (1989). "7 +/- 2 criteria for assessing and comparing spatial data structures." In *Design and Implementation of Large Spatial Databases, Proceedings of the First Symposium SSD '89*. (Buchman, A., O. Gunther, T. Smith, and Y. Wang, eds.). Berlin: Springer-Verlag. pp. 3-27.

Olsen, A.R., D.L. Stevens, and D. White (1998), Application of global grids in environmental sampling." *Computing Science and Statistics*, 30, ed. S. Weisberg, Interface Foundation of North America, Inc., Farfax Station, VA., this volume.

Rothman, D.H., and S. Zaleski (1997). *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics*, Cambridge: Cambridge University Press.

Sadourny, R., A. Arakawa, and Y. Mintz (1968). "Integration of the nondivergent barotropic vorticity equation with an icosahedral-hexagonal grid for the sphere." *Monthly Weather Review* 96(6): 351-356.

Samet, H. (1990). *The Design and Analysis of Spatial Data Structures*. Menlo Park, CA: Addison-Wesley.

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

Song, L. (1997). *Small Circle Subdivision Method for Development of Global Sampling Grid*. Unpublished Masters Thesis, Oregon State University, 176 pp.

Thuburn, J. (1997). "A PV-based shallow-water model on a hexagonal-icosahedral grid." *Monthly Weather Review* 125: 2328-2347.

White, D., A. J. Kimerling and W. S. Overton (1992). "Cartographic and geometric components of a global sampling design for environmental monitoring." *Cartography and Geographic Information Systems* 19(1): 5-22.

White D., A.J. Kimerling, K. Sahr, and L. Song (in press). "Comparing area and shape distortion on polyhedral-based recursive partitions of the sphere." *International Journal of Geographical Information Systems*.

Wickman, F. E., E. Elvers, and K. Edvarson (1974). "A system of domains for global sampling problems." *Geografiska Annaler* 56(3/4): 201-212.

Williamson, D. L. (1968). "Integration of the barotropic vorticity equation on a spherical geodesic grid." *Tellus* 20(4): 642-653.