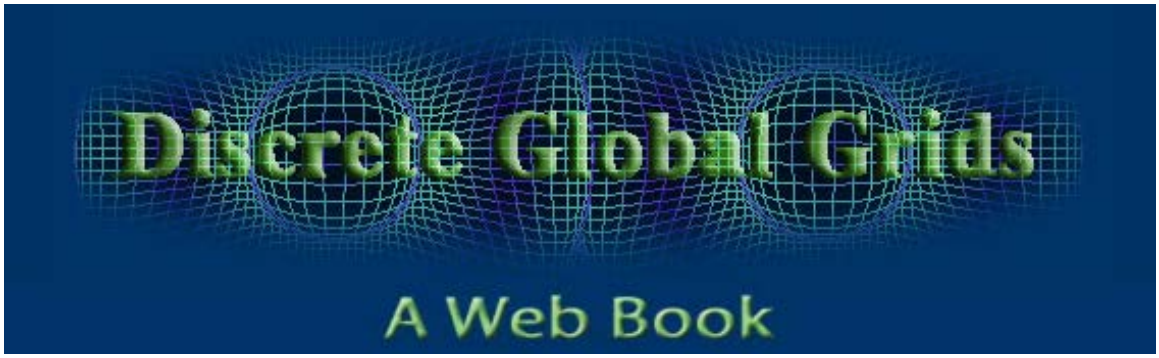


Discrete Global Grids: A Web Book (2002) was edited by Michael F. Goodchild and A. Jon Kimerling and produced with the support of the National Center for Geographic Information and Analysis (NCGIA). This document reproduces the eight 'chapters' of the book, each of which deals with a specific aspect of discrete global grids as a method for specifying geographic location on the surface of the earth. The chapters are as follows:

Chapter 1:

Developing an Equal Area Global Grid

by Small Circle Subdivision

by Lian Song, A. Jon Kimerling and Kevin Sahr

Department of Geosciences, Oregon State University

Corvallis, OR 97331

ABSTRACT. Modern environmental monitoring and modeling requires partitioning the earth's surface into a global grid optimized for survey sampling and unbiased, spatially complete data collection of relevant environmental phenomena. Prime requirements are that cells comprising the grid be equal in area, regular in shape, and highly compact on the earth's ellipsoidal surface. No existing global grid fully meets these criteria.

We propose a new equal area global partitioning method based upon small circle edges on the earth's surface, which we call the "small circle subdivision method." A detailed description of this method is presented, including its mathematical derivation and geometrical comparison with alternative methods. The small circle method appears to be the best developed to date to satisfy the essential criteria for a global grid.

ACKNOWLEDGEMENTS.

We acknowledge support from cooperative agreement CR 821672 between the US Environmental Protection Agency and Oregon State University. We also thank Denis White for reviewing the manuscript and preparing Figures 20-22.

Introduction

Monitoring the status of the global environment and assessing the condition of ecological resources across the earth have emerged as major scientific activities for the new century. Environmental phenomena and problems operating at different spatial and temporal scales, biogeochemical cycles and climate change being but two examples, are now being studied as global systems. A key issue is how to integrate both spatially and temporally disparate global data now being collected and made publicly available to the scientific community (Brooks 1981; Kahn and Braverman 1999).

One approach to data integration is to partition the earth into sampling or analysis units that form a hierarchy. Partitioning based on a hierarchy of political divisions, such as nations, states and counties, is a familiar approach linked to governmental data collection and dissemination efforts. However, scientists often favor developing a hierarchical, geometrically regular global partitioning system that is unbiased with respect to spatial patterns created by natural and human processes (White et al. 1998). A "flat earth" example would be to divide the flat surface into identical grid squares, each of which is then partitioned into quarters. This is called a four-fold partition (**Figure 1** left), whereas partitioning an initial triangle into nine identical triangles produces a nine-fold partition (**Figure 1** right). The grid cell hierarchy is formed by successive partitionings that we call recursion levels.

The geometrically ideal global partitioning system would consist of grid cells equal in surface area and identical in shape, akin to the square, equilateral triangular, or regular hexagonal grid cells that tessellate a flat surface. The centerpoints of these cells form geometrically uniform rectangular and triangular point sampling grids advantageous in statistical sampling designs for local areas. On the entire globe, however, area and shape regularity can be achieved only by projecting the faces of a Platonic polyhedron (tetrahedron, hexahedron, octahedron, dodecahedron, or icosahedron) onto the spherical or ellipsoidal

approximations to the earth's true shape. Since further partitioning of a spherical or ellipsoidal polyhedron face introduces unavoidable variation in cell area, shape, or both, grid cell optimization with respect to area, shape, and other criteria is a fundamental issue (Kimerling et al. 1999).

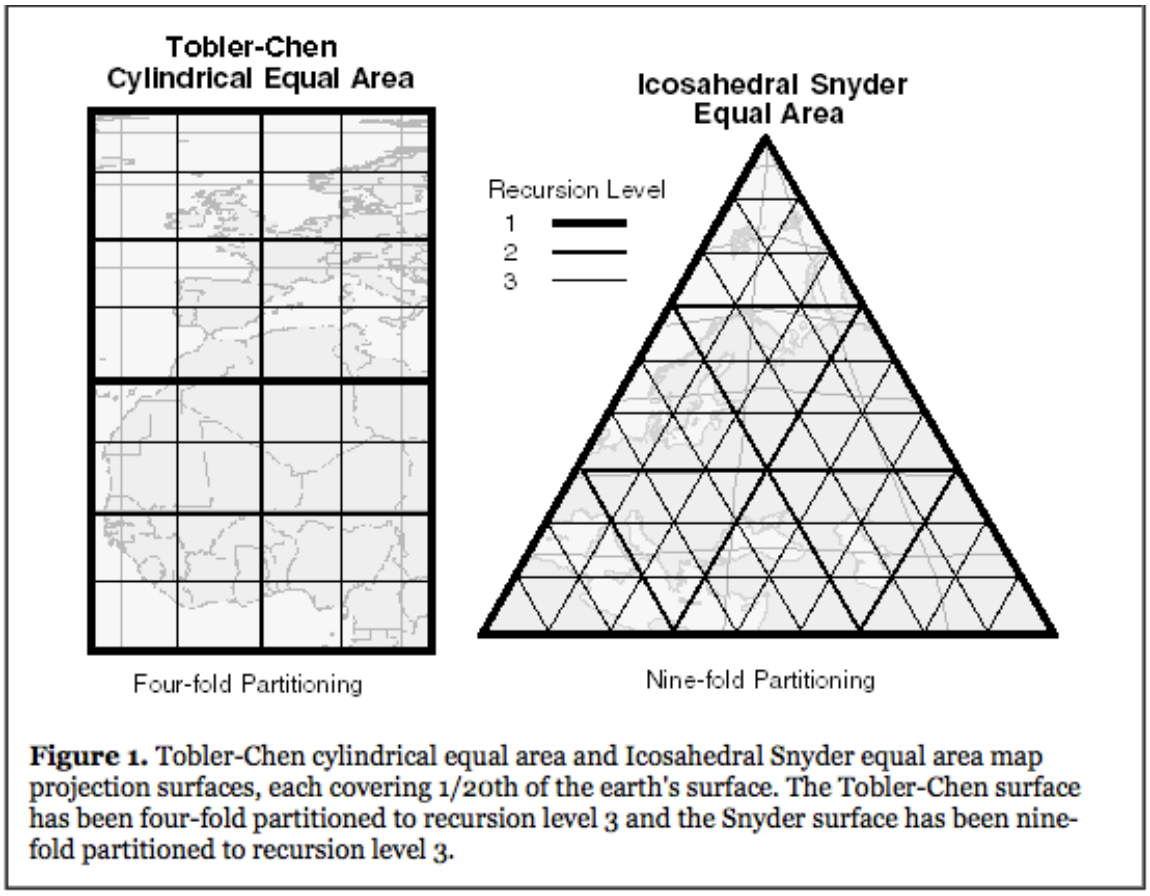


Figure 1. Tobler-Chen cylindrical equal area and Icosahedral Snyder equal area map projection surfaces, each covering 1/20th of the earth's surface. The Tobler-Chen surface has been four-fold partitioned to recursion level 3 and the Snyder surface has been nine-fold partitioned to recursion level 3.

Numerous approaches to partitioning the globe into areal grid cells have been proposed, and both Dutton (1999) and Kimerling et al. (1999) have grouped partitioning schemes into a small number of general categories. At the broadest level, partitioning methods can be divided into a) those that directly subdivide the sphere or ellipsoid into grid cells and b) those that project the globe onto one or more map projection surfaces which are partitioned in

a geometrically regular manner and then back-projected to the globe.

Equal area grid cells are a primary requirement to many scientists and spatial statisticians. Of the map projection based partitioning methods, the equal area polyhedral projections derived by Snyder (1992) satisfy this need (Figure 1 right), as does the cylindrical equal area projection approach (Figure 1 left) of Tobler and Chen (1986). Both introduce unavoidable variation in grid cell shape and compactness, the latter method suffering far greater in this respect. Directly partitioning the sphere into equal area quadrilateral cells has been achieved by mathematicians (Rakhnanov et al.1994) for a small number of cell densities, at the cost of inherently wide variations in cell shape characteristic of "constant area" global partitions (Kahn and Braverman 1999). Similar geometrical irregularities in point spacing have been noted by mathematicians studying the topologically dual problem of packing points on a sphere optimally (Saff and Kuijlaars 1997: Coexeter 1962). Directly partitioning the faces of Platonic polyhedra cannot achieve the equal area criterion when each face is divided into equal area sub-triangles and the three edges of each triangle are geodesic (great circle) lines defined by the laws of spherical trigonometry (Baumgardner and Frederickson. 1985).

To better satisfy the grid evaluation criteria, a new global partitioning method, the **small circle subdivision method**, is proposed. The small circle subdivision method, based upon the direct spherical subdivision approach, partitions the sphere or ellipsoid with small circles into sub-triangles equal in area that vary only slightly in shape. We first examine in detail the mathematical basis of the small circle subdivision method applied to a spherical icosahedron face, and then present results from implementing the method over a wide range of cell sizes. Results are presented in the form of a comparison with another proposed equal area global partitioning method based on the Icosahedral Snyder Equal Area (ISEA) map projection.

Small Circle Subdivision Method

The small circle subdivision method directly partitions the surface of the earth with small circles into equal area sub-regions. The small circle subdivision method can eliminate the inflexibility of great circle subdivision, because only one great circle is available between two points on a sphere (that is, the curvature of the great circle is fixed). Small circles can be any circles on a sphere with an arbitrary radius and axis. The curvature of a small circle through two points on a sphere can be flexibly changed. Thus, it is always possible to partition a triangle into equal area sub-triangles if appropriate small circles are computed.

Before describing the method of small-circle subdivision, terms that will be used frequently in the section must be defined. A circle on a sphere is an intersection of a sphere and a plane. A **small circle** is defined as any circle on a sphere. If the center of a small circle coincides with the center of the earth, we call it a **great circle**. Obviously, great circles are a subset of small circles. The **axis** of a small circle is the straight line that passes through the earth's center and is perpendicular to the plane of the small circle. The **pole** of a small circle is the point at which its axis intersects the surface of the earth (there are two intersection points, but we only consider the point close to the small circle plane as its pole). There exists a great circle parallel to the plane of a small circle, which we call the **relative equator** of the small circle. We define the **relative latitude** of a small circle as the angle between the plane of its relative equator and the straight line which passes through the sphere center and any point on the small circle. The **relative co-latitude** of a small circle is the angle between the axis of the small circle and the straight line through the sphere center and any point on the small circle. **Figure 2** shows a small circle cbd, its axis AA¢, its pole P, its relative equator CBD, its relative latitude BOb and its relative co-latitude bOP. Obviously, the axis, pole, equator and geodetic latitude in the traditional spherical coordinate system are particular examples of the above definitions. We also define a **small circle triangle** on a sphere as a region that is on the sphere and confined by three small circles (**Figure 3**). If all three circles are great circles, the triangle is called a **spherical triangle**. Obviously, spherical triangles are a subset of small circle triangles.

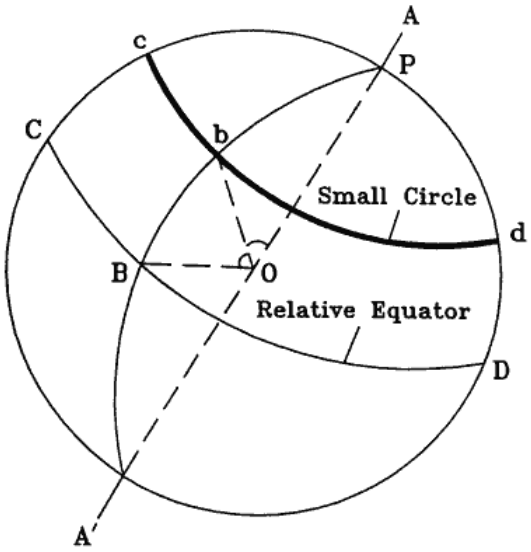


Figure 2. A small circle on a sphere with axis AA¢, pole P, relative equator CBD, relative latitude ∠BOb and relative co-latitude ∠bOP.

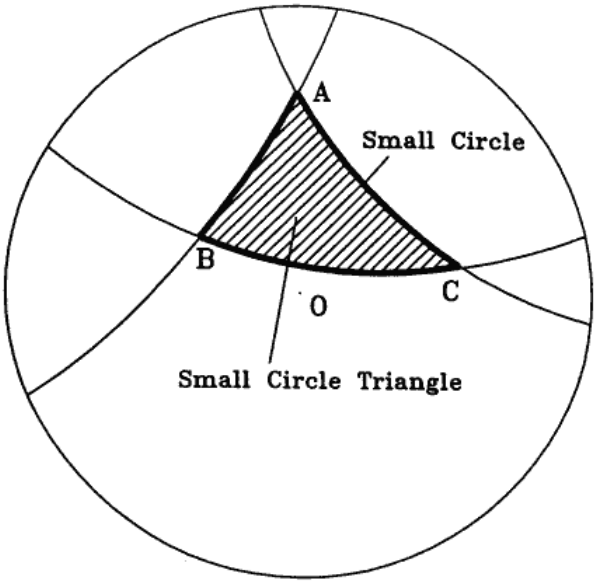


Figure 3. A small circle triangle on a sphere.

Our goal is to partition a particular spherical triangle, an **icosahedron face**, into equal area **sub-triangles**. The edges of the sub-triangles could be either small or great circles. We will consider two ways of dividing a triangle: four-fold and nine-fold subdivision. In the four-fold subdivision method, the three edges of a triangle are bisected and the bisection points are connected with small circles, producing four sub-triangles within the triangle. In the nine-fold subdivision method, the three edges of a triangle are trisected, and the trisection points as well as an inner center point are connected with one another by small circles, producing nine sub-triangles within the triangle. We define the initial spherical triangle (e.g. the icosahedron face) as a **recursion level** 0 triangle. If a recursion level 0 triangle is partitioned, we define the process as a recursion level 1 process and the resulting sub-triangles as recursion level 1 triangles. If the recursion level 1 triangles are partitioned, we define the process as a recursion level 2 process and the resulting sub-triangles as recursion level 2 triangles, and so on.

Four-Fold Subdivision Method

In the four-fold subdivision method, the three edges of a triangle are bisected and the bisection points are connected with small circles, producing four sub-triangles within the parent triangle. If the bisection points were connected with great circles, the four resulting sub-triangles would not have equal areas because only one great circle passing through two points on a sphere is available, that is, the curvature of the great circle is fixed. For example, if we partition an icosahedron face, the great circle segment between two bisection points always bends outward from the center of the parent triangle, resulting in a larger area of the middle sub-triangle (**Figure 4**). If the bisection points are connected with small circles, we could obtain four sub-triangles equal in area, because there exist infinite numbers of small circles between two points on a sphere and we can arbitrarily change the radius and axis (or curvature) of a small circle by moving its pole up or down on the sphere. In **Figure 4**, if we move the pole of a small circle toward the center sub-triangle, the radius of the small circle will become smaller

and the arc of the small circle segment will shift toward the center of the parent triangle, resulting in a smaller area for the center sub-triangle and a larger area for the outer sub-triangle associated with the small circle segment. The area of the outer sub-triangle is determined only by its associated small circle segment, independent of the other two small circle segments. If the pole of the small circle is moved to a position on the sphere such that the area of the outer sub-triangle is just one-fourth of the area of its parent triangle, the small circle segment is determined. The other two small circle segments can be determined in the same way. Because each of the four sub-triangles has an area one-fourth of the parent triangle, they obviously are equal in area.

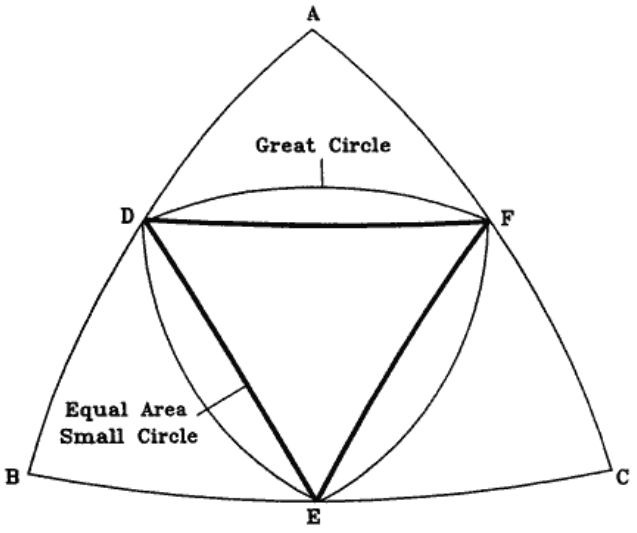


Figure 4. Four-fold small circle and great circle subdivision geometry for an icosahedron face.

Mathematical Derivations

We now derive mathematical equations and determine numerical methods to implement the subdivision process discussed above, so that the following essential information

about sub-triangles at each recursion level will be obtained: (1) the latitudes and longitudes of all vertices; (2) the relative co-latitudes of all edges; (3) the geodetic latitudes and longitudes of poles for all edges; (4) the arc lengths of all edges; and (5) the area of all sub-triangles.

For simplicity, we represent the earth as a unit sphere, so that the arc length (radian measure) of a great circle segment between two points on the sphere is the same as the angle it subtends. For a sphere with radius r, linear measurements are scaled by a factor of r and area measurements by r^2.

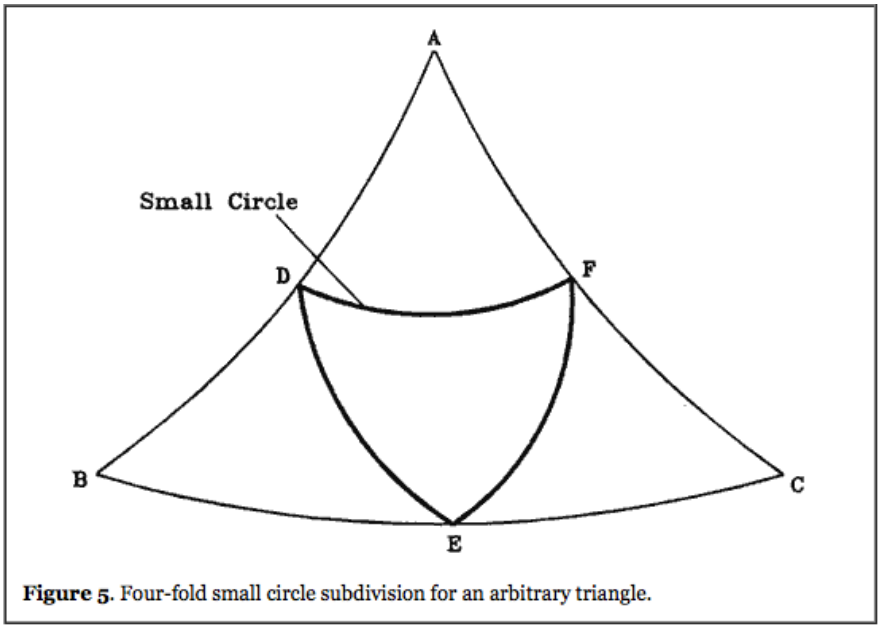


Figure 5. Four-fold small circle subdivision for an arbitrary triangle.

Consider an arbitrary initial triangle ABC on a unit sphere in **Figure 5**, with its three edges made of either small or great circles. We assume that the following information about the triangle is known a priori:

- (1) the latitude and longitude of each vertex;
- (2) the latitude and longitude of the pole for each side;
- (3) the arc length of each side (small circle segment);
- (4) the relative co-latitude of each side.

- (5) the area of the triangle.

Our problem is to find three small circles connecting each pair of the midpoints D, E and F, so that four resulting sub-triangles will have equal areas. We first find the coordinates of midpoints D, E and F. Consider, in particular, the edge AB, with its pole P_{AB} at $(\theta_{P_{AB}}, \phi_{P_{AB}})$ and its relative co-latitude aAB (**Figure 6**). Denote the spherical coordinates for the vertices A and B as (θ_A, ϕ_A) and (θ_B, ϕ_B), respectively. Construct a Cartesian coordinate system with the origin located at the center of the earth, the x-y plane on the equator and the z-axis pointing to the north pole. Standard formulas for transforming from spherical coordinates to Cartesian coordinates of a point on the unit sphere are:

(1)

$$x = \cos(\theta)\cos(\phi)$$

$$y = \cos(\theta)\sin(\phi)$$

$$z = \sin(\theta)$$

The Cartesian coordinates, (x_A, y_A, z_A), (x_B, y_B, z_B) and $(x_{P_{AB}}, y_{P_{AB}}, z_{P_{AB}})$ for points A, B, and P_{AB}, respectively, can be computed using **equations (1)**. Thus, the Cartesian coordinates of the midpoint D¢ of the chord between the point A and the point B are:

(2)

$$x_{D'} = (x_A + x_B)/2$$
$$y_{D'} = (y_A + y_B)/2$$
$$z_{D'} = (z_A + z_B)/2$$

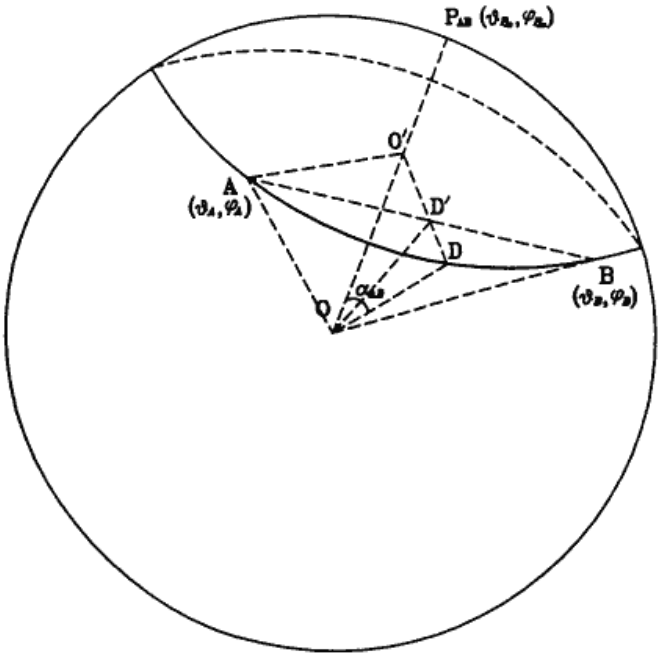


Figure 6. Geometry of a small circle midpoint calculation.

We denote the vector from the earth center O to the point O¢, the center of the small circle, as $\vec{O}'$, and the vector from the earth center O to the pole P_{AB} of the small circle as $\vec{P}_{AB}$. So we get:

(3)

$$\vec{O}' = \vec{P}_{AB} \cos \alpha_{AB}$$

or

(4)

$$x_{O'} = x_{P_{AB}} \cos \alpha_{AB}$$
$$y_{O'} = y_{P_{AB}} \cos \alpha_{AB}$$
$$z_{O'} = z_{P_{AB}} \cos \alpha_{AB}$$

The vector $\vec{D}$ which points from the earth center O to the midpoint D of the small circle segment AB is given by:

(5)

$$\vec{D} = \vec{O'} + \frac{\vec{D'} - \vec{O'}}{\left|\vec{D'} - \vec{O'}\right|} \sin \alpha_{AB}$$

where $\frac{\vec{D'} - \vec{O'}}{\left|\vec{D'} - \vec{O'}\right|}$ is a unit vector pointing from the center O¢ of the small circle to the point D¢ or D, and $\sin\alpha_{AB}$ is the radius of the small circle. Transforming **equation (5)** from vector expression to scalar expressions, we get the Cartesian coordinates of the midpoint D:

(6)

$$x_D = x_{O'} + \frac{(x_{D'} - x_{O'}) \sin \alpha_{AB}}{\sqrt{(x_{D'} - x_{O'})^2 + (y_{D'} - y_{O'})^2 + (z_{D'} - z_{O'})^2}}$$

$$y_D = y_{O'} + \frac{(y_{D'} - y_{O'}) \sin \alpha_{AB}}{\sqrt{(x_{D'} - x_{O'})^2 + (y_{D'} - y_{O'})^2 + (z_{D'} - z_{O'})^2}}$$

$$z_D = z_{O'} + \frac{(z_{D'} - z_{O'}) \sin \alpha_{AB}}{\sqrt{(x_{D'} - x_{O'})^2 + (y_{D'} - y_{O'})^2 + (z_{D'} - z_{O'})^2}}$$

The spherical coordinates of the midpoint D are calculated using the following equations:

(7)

$$\theta = \arcsin(z)$$

$$\phi = \arctan(\frac{y}{x})$$

The other two midpoints, E and F, can be found in the same manner.

After three midpoints have been found, we next determine three unknown small circles DE, EF and DF in **Figure 5**. First consider the small circle DF and the sub-triangle ADF. We will find the small circle DF so that the area of sub-triangle ADF is one-fourth of the area of the parent triangle ABC. We have shown that the curvature of a small circle depends both on the position of its pole and on its relative co-latitude. But for a small circle crossing two fixed points on a sphere (i.e., the geodetic coordinates of the two points are known), its relative co-latitude will be the only variable in determining the small circle because its pole can be found from the relative co-latitude. To find the relative latitude of the small circle, we connect two midpoints, D and F, with a great circle (**Figure 7**). Great circle segment DGF and small circle segment DSF bound the shaded area denoted by SL. We call it a **semi-lune** (in spherical trigonometry a lune is a region on a sphere bounded by two great circle segments). Because the small circle segments AD and AF and the great circle segment DGF are known, the area of the triangle ADGF confined by the three segments can be computed. Thus, the area of the semi-lune SL, denoted by S_{SL} , is calculated as the area difference between the triangle ADGF and the triangle ADSF that is one-fourth the area of the parent triangle. To find the area of the triangle ADGF, construct a spherical triangle AGDGFGA (**Figure 8**) by connecting each pair of the points A, D and F with a great circle. The area of triangle ADGF (i.e. the triangle ASDGFSA in **Figure 8**) is the area difference or sum of the spherical triangle AGDGFGA and two semi-lunes (SL_1 and SL_2) depending on the curvature of the small circles ASD and ASF. For convenience, we assume that the area of a semi-lune has a positive value if its small circle edge is outside its great circle edge,

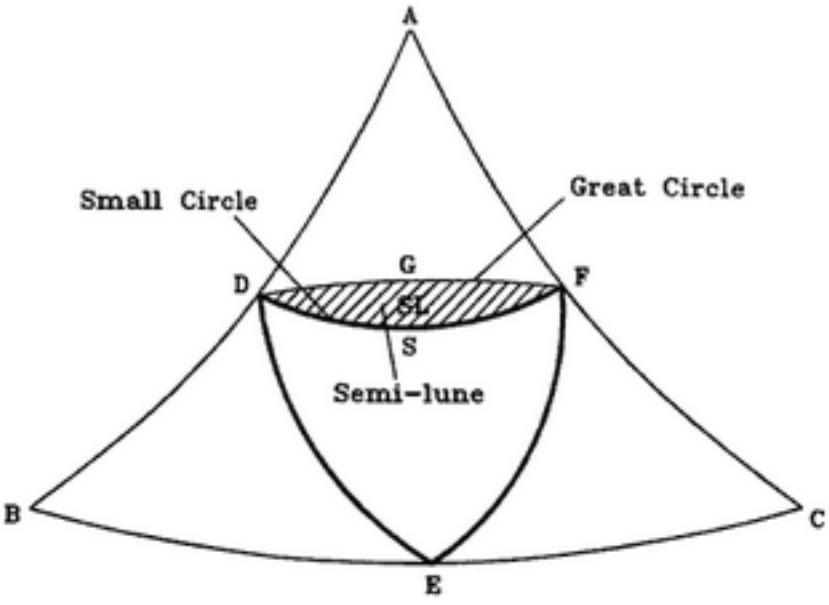


Figure 7. A semi-lune confined by a great circle and a small circle.

and a negative value otherwise. In the **Figure 8**, for example, the semi-lunes SL_1 and SL_2 have negative areas and the semi-lune SL has a positive area. So the area of the semi-lune SL can be written as:

(8)

$$S_{SL} = \frac{1}{4} S_{abc} - S_{ADF} - S_{SL_1} - S_{SL_2}$$

where S_{abc} is the area of the parent triangle ABC, S_{ADF} is the area of the spherical triangle ADF (i.e. the spherical triangle AGDGFGA in **Figure 8**), S_{SL_1} and S_{SL_2} are areas of two semi-lunes SL_1 and SL_2, respectively. Our problem simplifies to finding the relative co-latitude of the small circle DSF

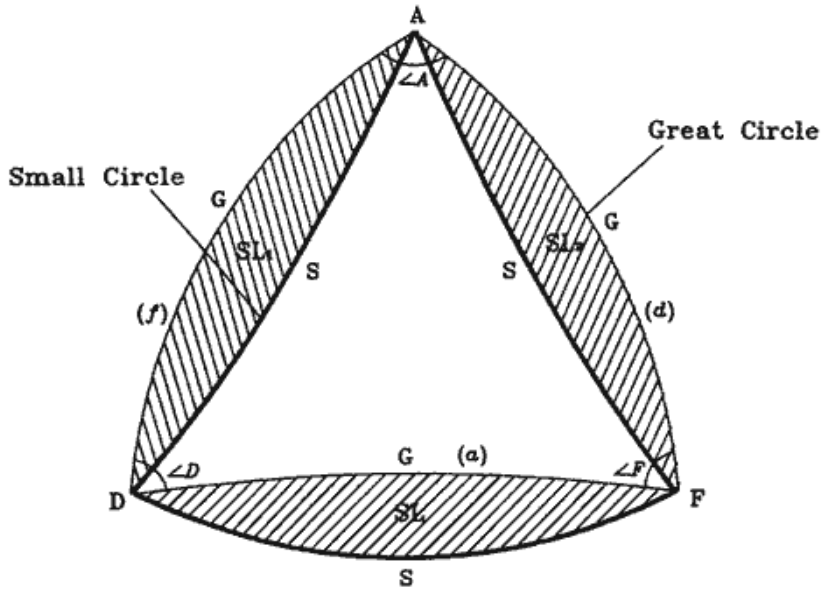


Figure 8. Geometry of area representation of the unknown semi-lune SL by areas of two known semi-lunes and one spherical triangle. Heavy lines represent small circles and light lines represent great circles.

so that the area of the semi-lune SL satisfies **equation (8)**. The following steps are needed to calculate the area of the semi-lune, S_{SL} :

(1)

Compute the area, S_{ADF}, of the spherical triangle ADF. The arc lengths of great circle segments AGD, DGF and AGF denoted by f, a and d, respectively, can be obtained by calculating the spherical distances between pairs of vertices. The arc length f of great circle segment AGD, for example, is:

$$\cos(f) = \sin(\theta_A)\sin(\theta_D) + \cos(\theta_A)\cos(\theta_D)\cos(\phi_A - \phi_D)$$

or

(9)

$$f = \arccos[\sin(\theta_A)\sin(\theta_D) + \cos(\theta_A)\cos(\theta_D)\cos(\phi_A - \phi_D)]$$

where θ_A and θ_D are the latitudes of the vertices A and D, respectively, and ϕ_A and ϕ_D are the longitudes of the vertices A and D. Similarly, we can obtain the arc lengths a and d of the great circle segments DGF and AGF. Denote the vertex angles of the spherical triangle by ĐA, ĐD and ĐF. The law of cosines for sides of a spherical triangle gives:

(10)

$$\cos(a) = \cos(d)\cos(f) + \sin(d)\sin(f)\cos(\angle A)$$

So the vertex angle ĐA is:

(11)

$$\angle A = \arccos\left(\frac{\cos(a) - \cos(d)\cos(f)}{\sin(d)\sin(f)}\right)$$

Similarly, the vertex angles, ĐD and ĐF, can be obtained. The area of the spherical triangle ADF is the spherical excess (sum of the interior angles minus p), or:

(12)

$$S_{ADF} = \angle A + \angle D + \angle F - \pi$$

(2)

Compute the areas S_{SL_1} and S_{SL_2} of two known semi-lunes SL_1 and SL_2. First consider semi-lune SL_1 (**Figure 9**). The pole, P_{AD}, of the small circle ASD is located at $(\theta_{P_{AD}}, \phi_{P_{AD}})$ and its relative co-latitude is a_{AD}. Connect points A and D with great circles to the pole P_{AD} and calculate the area of the spherical triangle P_{AD}AGD. The area

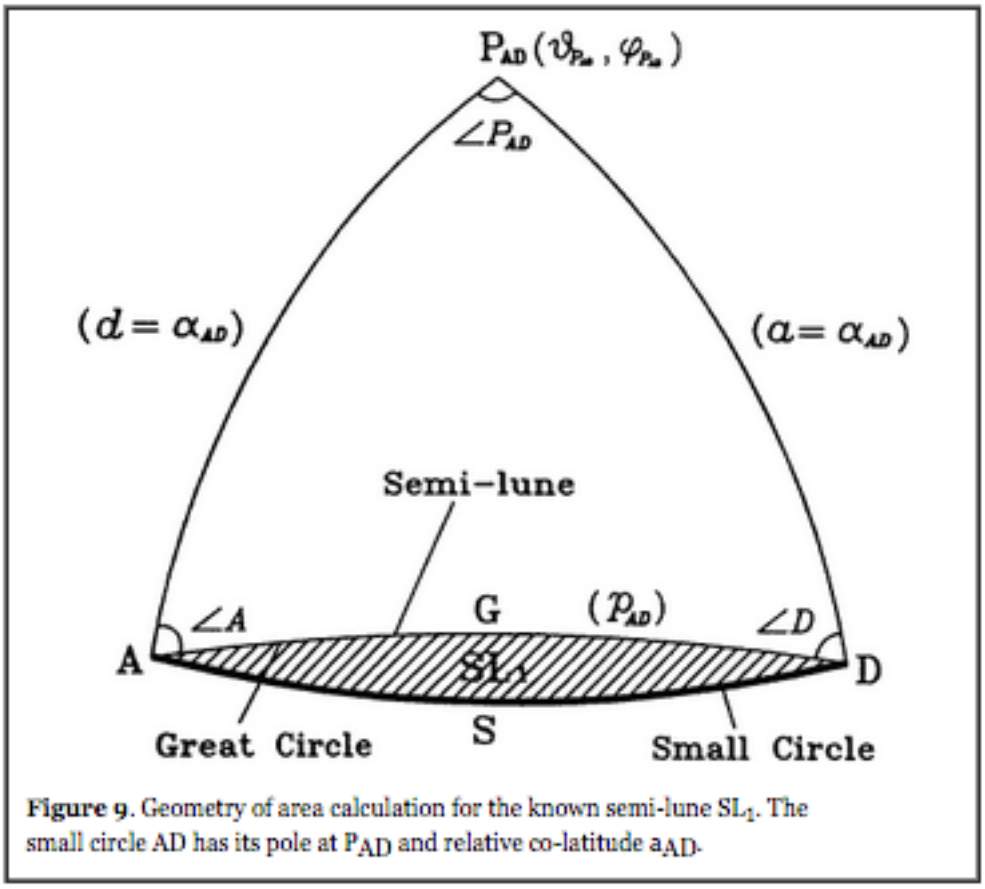


Figure 9. Geometry of area calculation for the known semi-lune SL_1. The small circle AD has its pole at P_{AD} and relative co-latitude a_{AD}.

calculation is similar to that above, i.e., first compute the vertex angles of the spherical triangle, ĐP$_{AD}$, ĐA and ĐD, and then calculate the area:

(13)

$$S_{P_{AD}AD} = \angle P_{AD} + \angle A + \angle D - \pi$$

The area of the small circle triangle P$_{AD}$ASD (polar wedge) is given by:

(14)

$$S_{p_{AD}ad} = \angle P_{AD}(1 - \cos \alpha_{AD})$$

Then the area of the semi-lune SL_1 is:

$$S_{SL_1} = S_{p_{AD}ad} - S_{P_{AD}AD}$$

or

(15)

$$S_{SL_1} = \angle P_{AD}(1-\cos\alpha_{AD}) - (\angle P_{AD} + \angle A + \angle D - \pi)$$

Because the vertex angle ÐA is equal to ÐD, we get

(16)

$$S_{SL_1} = \pi - 2\angle A - \angle P_{AD}\cos\alpha_{AD}$$

Now we need to find the angles ÐA and $ÐP_{AD}$. We know that the arc length d (or a) of the great circle segment $P_{AD}GA$ (or $P_{AD}GD$) is the relative co-latitude, a_{AD}, of the small circle ASD. The arc length, p_{AD}, of the great circle segment AGD is the spherical distance from point A to point D, which can be calculated using **equation (9)**. The law of cosines for sides of a spherical triangle gives:

(17)

$$\cos p_{AD} = \cos d\cos a + \sin d\sin a\cos\angle P_{AD}$$

Because *d*=*a*=a_{AD}, **equation (17)** becomes:

$$\cos p_{AD} = \cos^2\alpha_{AD} + \sin^2\alpha_{AD}\cos\angle P_{AD}$$

(18)

$$= (1-\sin^2\alpha_{AD}) + \sin^2\alpha_{AD}\cos\angle P_{AD}$$

So the vertex angle $ÐP_{AD}$ is:

(19)

$$\angle P_{AD} = \arccos(\frac{\cos p_{AD} - 1}{\sin^2 \alpha_{AD}} + 1)$$

Now calculate the vertex angle ÐA. The law of cosines for angles of a spherical triangle gives:

$$\cos \angle P_{AD} = -\cos^2 \angle A + \sin^2 \angle A \cos p_{AD}$$

(20)

$$= \sin^2 \angle A - 1 + \sin^2 \angle A \cos p_{AD}$$

A manipulation of **equation (20)** gives the vertex angle A:

(21)

$$\angle A = \arcsin\left(\sqrt{\frac{1 + \cos \angle P_{AD}}{1 + \cos p_{AD}}}\right)$$

Substituting **equations (19)** and **(21)** into **(16)**, the area of semi-lune SL_1 is computed. Similarly, we can compute the area of semi-lune SL_2.

(3)

Find the area, S_{SL}, of the unknown semi-lune SL. After the area, S_{ADF}, of the spherical triangle ADF and the areas, S_{SL_1} and S_{SL_2}, of semi-lunes SL_1 and SL_2 are found, the area of the semi-lune SL can be computed from **equation (8)**.

Next, determine the small circle DSF in **Figure 7**. As mentioned above, if we find the relative co-latitude of the small circle, the small circle will be determined. Denote the relative co-latitude of the small circle as a_{DF} and its pole as P_{DF}. Similar to the area calculation above for known semi-lune SL_1, the area of the semi-lune SL can be expressed by a_{DF} as follows:

(22)

$$S_{SL} = \pi - 2\angle D - \angle P_{DF} \cos \alpha_{DF}$$

$$\angle P_{DF} = \arccos\left(\frac{\cos p_{DF} - 1}{\sin^2 \alpha_{DF}} + 1\right)$$

and

$$\angle D = \arcsin\left(\sqrt{\frac{1 + \cos \angle P_{DF}}{1 + \cos p_{DF}}}\right)$$

where p_{DF} is the spherical distance from the point D to the point F, which is computed using equation (9). Angles $ÐP_{DF}$ and ÐD are vertex angles of the spherical triangle P_{DF}DGF (Figure 10). There are three equations and three unknowns (a_{DF}, $ÐP_{DF}$ andÐD), so a_{DF} can be determined. It is very difficult to get an analytical solution from these equations because the set of equations is nonlinear. So we turn to numerical methods to get an approximate solution with prescribed precision. We assume an initial value of $a_{DF}$and find the area of the semi-lune by solving **equation (22)**. Then we adjust the value ofa_{DF} up or down by comparing the computed area with the desired amount obtained above. If the computed area is larger or smaller than the desired amount, the value of $a_{DF}$is increased or decreased, respectively. If the difference between the computed area and the desired amount is less than a prescribed precision, we stop computation and select the value of a_{DF} as the final relative co-latitude of

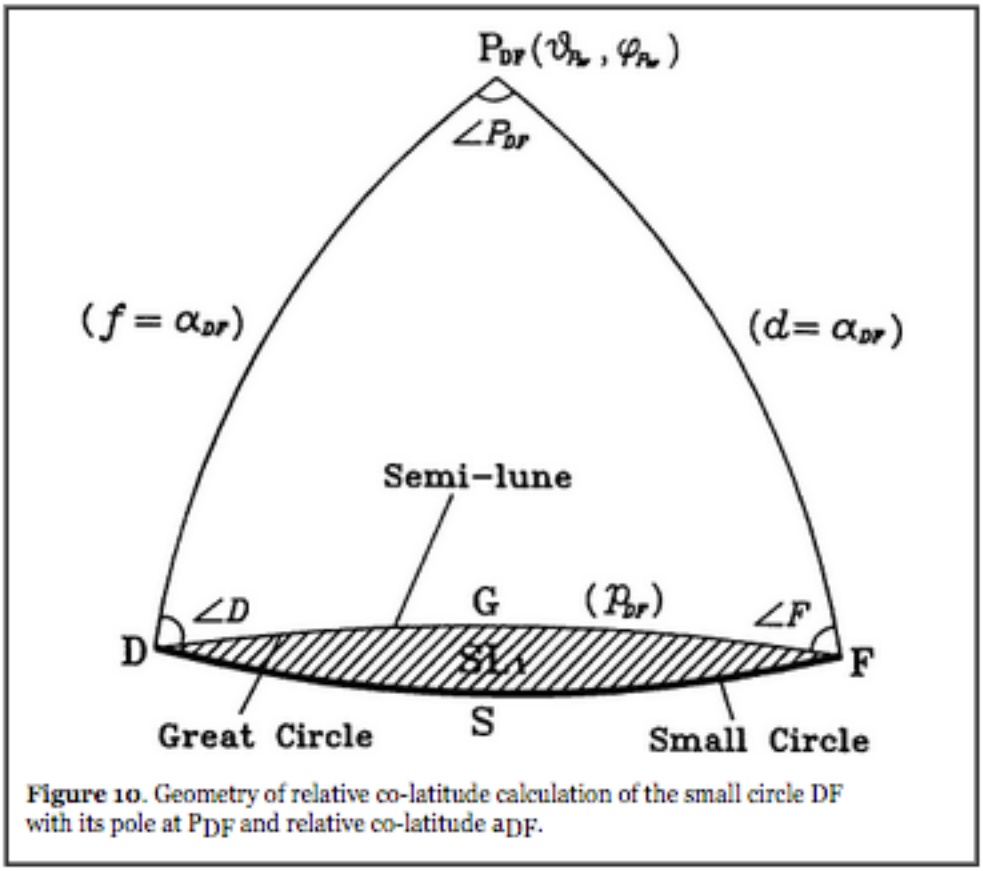


Figure 10. Geometry of relative co-latitude calculation of the small circle DF with its pole at P_{DF} and relative co-latitude a_{DF}.

small circle DSF. To speed up the convergence, we will apply the binary-tree method. We first find the range of the relative co-latitude, and then determine the relative co-latitude iteratively using bisection beginning with the middle point of the range. Because we always use the pole which is close to the small circle plane, the maximum value of the relative co-latitude is p/2. If we are only interested in the small circle segment DSF less than a semi-circle, the minimum value of the relative co-latitude is arcsin(t_{DF}/2) where t_{DF} is the chord length between D and F. This consideration is reasonable because we require that sub-triangles have similar shapes. If one of the small circle edges of a sub-triangle has an arc length more than a semi-circle, the shape of the small circle triangle will be unacceptably strange. This requirement sets a limitation on subdivision precision. In some cases where a triangle has an unusual shape and the prescribed precision is high, a small circle segment larger than a semi-circle may be needed

to subdivide the triangle. In such cases, if we search the value of a_{DF} only in the range (arcsin(t_{DF}/2),p/2), the computation would loop forever and the small circle would never be found. In practice, this situation has never been encountered when partitioning an icosahedron face up to recursion level 9, but was encountered once when partitioning an octahedron face at recursion level 8.

After the relative co-latitude of the small circle DSF has been found, we next determine the spherical coordinates of the pole of the small circle, $(\theta_{P_{DF}}, \phi_{P_{DF}})$, in **Figure 11**. Denote the Cartesian coordinate of the pole P_{DF} by $(x_{P_{DF}}, y_{P_{DF}}, z_{P_{DF}})$. The pole P_{DF} is on the sphere:

(23)

$$x_{P_{DF}}^2 + y_{P_{DF}}^2 + z_{P_{DF}}^2 = 1$$

The Cartesian coordinates of the endpoints, D and F, denoted by (x_D, y_D, z_D) and (x_F, y_F, z_F), respectively, can be obtained from **equation (1)**. The angle between the vectors $(x_{P_{DF}}, y_{P_{DF}}, z_{P_{DF}})$ and (x_D, y_D, z_D) or (x_F, y_F, z_F) is the relative co-latitudea_{DF} of the small circle DF (**Figure 11**), so the inner products give:

(24)

$$\cos \alpha_{DF} = x_D x_{P_{DF}} + y_D y_{P_{DF}} + z_D z_{P_{DF}}$$
$$\cos \alpha_{DF} = x_F x_{P_{DF}} + y_F y_{P_{DF}} + z_F z_{P_{DF}}$$

Equations (23) and **(24)** are solved together for $x_{P_{DF}}$, $y_{P_{DF}}$ and $z_{P_{DF}}$. There are two possible solutions here. If the small circle segment DSF is outside the great circle segment

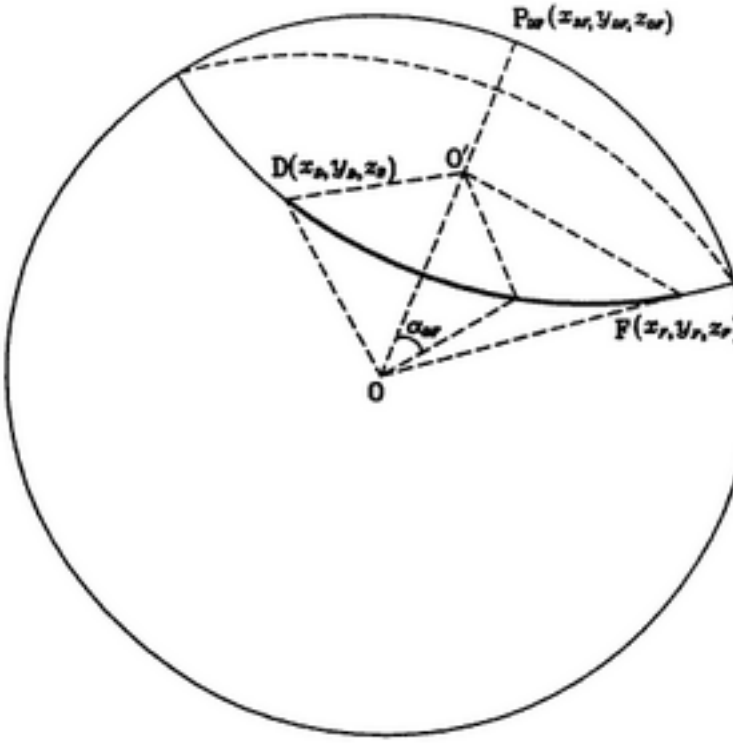


Figure 11. Geometry of pole determination of the small circle DF.

DGF, we choose one solution close to the vertex A. Otherwise, we choose the solution far from the vertex A. Then we transform the Cartesian coordinates back to spherical coordinates:

(25)

$$\theta_{P_{DF}} = \arcsin z_{P_{DF}}$$

$$\phi_{P_{DF}} = \arctan(y_{P_{DF}} / x_{P_{DF}})$$

The arc length of the small circle segment DSF can also be obtained from its relative co-latitude and the vertex angle $ÐP_{DF}$. The expression for the small circle arc length is:

(26)

$$l_{DF} = \angle P_{DF} \sin \alpha_{DF}$$

We determined the small circle DF in **Figure 5**. The other two small circles DE and EF can be obtained in the same way as above. Each of the four sub-triangles has an area equal to one-fourth the area of their parent triangle, so the triangle ABC is divided into four equal-area sub-triangles.

Nine-fold Subdivision Method

In the nine-fold method, all three edges of a triangle are trisected and the trisection points, as well as a central inner point, are connected by small circles, producing nine sub-triangles within the parent triangle. The basic problem is still to determine the small circles so that the resulting nine sub-triangles have equal areas. The method of small circle determination is similar to the four-fold case, with a few exceptions. First, the central inner point of a triangle needs to be found and to be connected with six trisection points of three sides with small circles so as to construct nine sub-triangles (**Figure 12**). If six trisection points are directly connected with one another by small circles, three inner small circles, each dividing the hexagon DEFGHI into two equal-area quadrilaterals, may not intersect at one point. The position of the inner point will affect the compactness of the subdivision method. We have tried several ways to choose the inner point, and will illustrate these in detail later. Secondly, each of six inner sub-triangles has two unknown sides. This is different from the two-frequency method where each sub-triangle has only one unknown side. Thus, to determine six inner small

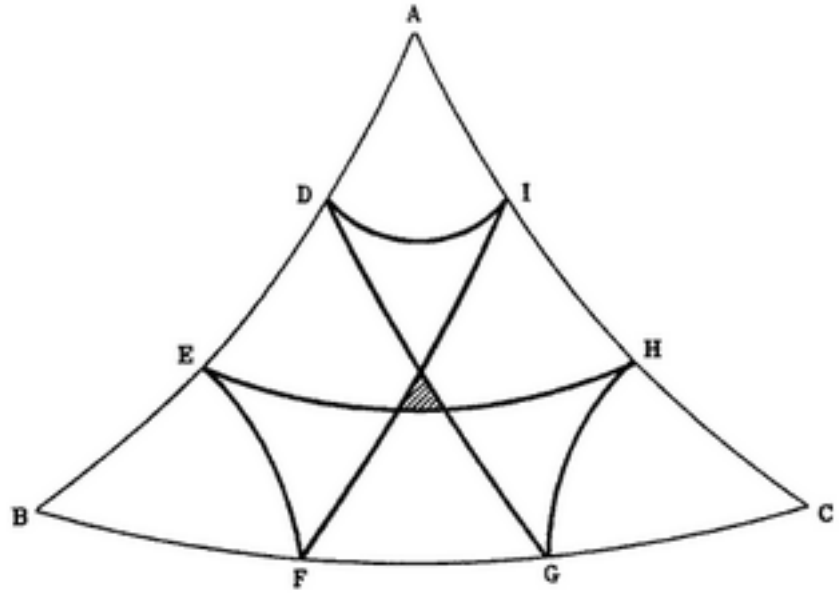


Figure 12. Small circles directly connect six trisecting points with one another. Three inner small circles (heavy lines), each of them dividing the hexagon DEFGHI into two equal-area quadrilaterals, do not intersect at one point and so an inner point is required.

circles in three-frequency subdivision method, at least one initial small circle should be known a priori to simplify computations. The others will be derived one by one from the initial small circles.

Consider the small circle triangle ABC in **Figure 13**. First, we need to find the trisection points for the three sides of the triangle. Consider, in particular, the edge AB, with its pole P_{AB} at $(\theta_{P_{AB}}, \phi_{P_{AB}})$ and its relative co-latitude a_{AB} (**Figure 14**). Denote the spherical coordinates for the vertices A and B as (θ_A, ϕ_A) and (θ_B, ϕ_B), respectively. The Cartesian coordinates, (x_A, y_A, z_A), (x_B, y_B, z_B) and $(x_{P_{AB}}, y_{P_{AB}}, z_{P_{AB}})$ for points A, B, and P_{AB} can be

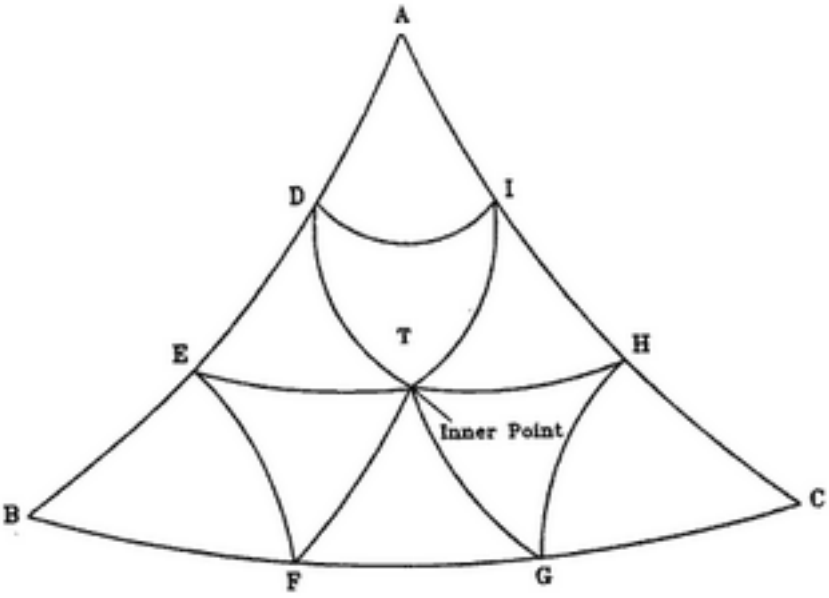


Figure 13. Nine-fold small circle subdivision for an arbitrary triangle ABC.

calculated from **equation (1)**. Assume that D and E are two trisection points of side AB. Because D and E are on the unit sphere, their Cartesian coordinates (x_D, y_D, z_D) and (x_E, y_E, z_E) satisfy:

(27)

$$x_D^2 + y_D^2 + z_D^2 = 1$$

(28)

$$x_E^2 + y_E^2 + z_E^2 = 1$$

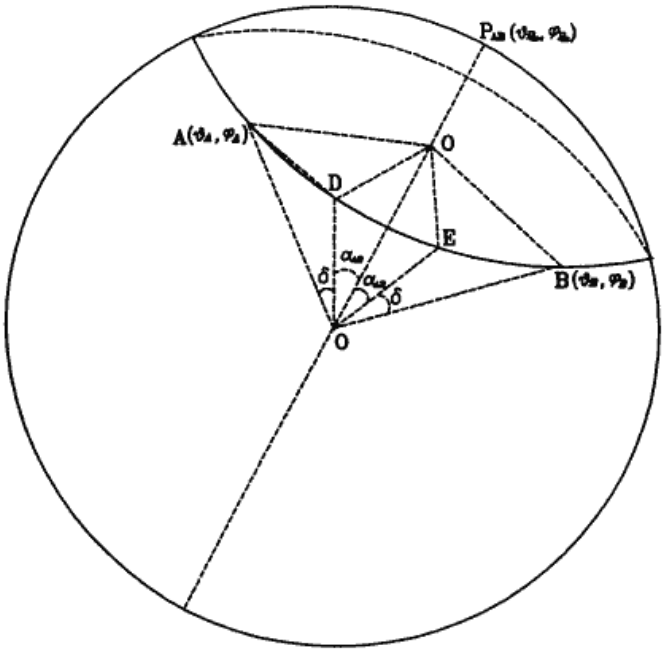


Figure 14. Geometry of trisecting point calculation of a small circle AB with its pole at P_{AB} and relative co-latitude a_{AB}.

We know that the angle between the vector $(x_{P_{AB}}, y_{P_{AB}}, z_{P_{AB}})$ and each of the vectors (x_D, y_D, z_D) and (x_E, y_E, z_E) is the relative co-latitude, a_{AB}, of the small circle AB, so the inner product gives:

(29)

$$x_{P_{AB}} x_D + y_{P_{AB}} y_D + z_{P_{AB}} z_D = \cos \alpha_{AB}$$

(30)

$$x_{P_{AB}} x_E + y_{P_{AB}} y_E + z_{P_{AB}} z_E = \cos \alpha_{AB}$$

Denote the angle between vectors (x_D, y_D, z_D) and (x_A, y_A, z_A) as d, so the inner product gives:

(31)

$$x_A x_D + y_A y_D + z_A z_D = \cos \delta$$

The angle between vectors (x_E, y_E, z_E) and (x_B, y_B, z_B) is also d, so we get:

(32)

$$x_B x_E + y_B y_E + z_B z_E = \cos \delta$$

We need to compute the angles d. Applying the plane law of cosines to the plane triangle connecting A, D and the earth center O (**Figure 14**), the chord length between A and D is:

(33)

$$t_{AD}{}^2 = 2 - 2\cos \delta$$

In the plane triangle connecting A, D and the center O¢ of the small circle AD, the length O¢A or O¢D is the radius of the small circle AD, which is $r=\sin a_{AB}$ (**Figure 14**). The angle ÐAO¢D is one-third of the angle ÐAO¢B which is equal to the arc length of the small circle segment AB, l_{AB}, divided by r. So the plane law of cosines gives:

$$t_{AD}{}^2 = r^2 + r^2 - 2r^2 \cos(\frac{l_{AB}}{3r})$$

or

(34)

$$t_{AD}{}^2 = 2\sin^2 \alpha_{AB} - 2\sin^2 \alpha_{AB} \cos(\frac{l_{AB}}{3\sin \alpha_{AB}})$$

Solving the **equations (33)** and **(34)** for cosd:

(35)

$$\cos\delta = 1 - \sin^2\alpha_{AB} + \sin^2\alpha_{AB}\cos(\frac{l_{AB}}{3\sin\alpha_{AB}})$$

Substituting into **equations (31)** and **(32)**, we get:

(36)

$$x_A x_D + y_A y_D + z_A z_D = 1 - \sin^2\alpha_{AB} + \sin^2\alpha_{AB}\cos(\frac{l_{AB}}{3\sin\alpha_{AB}})$$

(37)

$$x_B x_E + y_B y_E + z_B z_E = 1 - \sin^2\alpha_{AB} + \sin^2\alpha_{AB}\cos(\frac{l_{AB}}{3\sin\alpha_{AB}})$$

We must solve **equations (27)**, **(29)** and **(36)** for (x_D, y_D, z_D), and **equations (28)**, **(30)** and **(37)** for (x_E, y_E, z_E). Each has two solutions. We only select the solution close to the point B as the Cartesian coordinates for point D, and the solution close to the point A as the Cartesian coordinates for point E. The trisection points of other two sides can be computed in the same way.

After the trisection points for three sides of triangle ABC are found, we next determine the three outer small circles DI, FE and HG. The method to find the three small circles has been introduced previously in the four-fold subdivision method. The only difference is that each area of the three outer sub-triangles ADI, BFE and CHG is one-ninth the area of triangle ABC.

Next, we need to determine the inner point of triangle ABC. As mentioned previously, the determination of the inner point will affect the compactness of the subdivision method. So the position of the inner point M is dependent upon the shape of the hexagon DEFGHI which is confined by six small circles. The ideal way is to select the center point of the hexagon as the inner point. But the calculation of the center point is complex because it involves integration over the unit sphere bounded by six small circles. The other way to locate an inner point is to

find three small circles EH, GD and IF using the same method as in the four-fold case so that each of three triangles AEH, BGD and CIF has an area four-ninths the area of triangle ABC. Actually, each of the three small circles divides the hexagon DEFGHI into two equal-area quadrilaterals. These three small circles intersect at three points and define a small triangle (**see Figure 12**). We arbitrarily choose a point inside the small triangle as an inner point (e.g., the coordinates of the point could be the average coordinates of the three intersection points). However, a difficulty will be encountered later when we determine six inner small circles using the inner point. As we explain later, at least one initial small circle should be known a priori to find other small circles. But there is no proper way to obtain such a small circle if we use the inner point. To simplify our calculation, we use two other methods to find the inner point. First, find the small circle EH that divides the hexagon DEFGHI into two equal-area quadrilaterals, and then choose the bisection point of the small circle segment EH as an inner point (**Figure 15**). Second, find the small circles DG and IF in the same way as above, and then select the intersection point of the two small circles as an inner point (**Figure 16**). The Cartesian coordinates (x_T, y_T, z_T) of the intersection point T, can be calculated by the following steps:

(1)

Denote the relative co-latitude of small circle DG as a_{DG}, the spherical coordinates of its pole P_{DG} as $(\theta_{P_{DG}}, \phi_{P_{DG}})$, the relative co-latitude of small circle IF as a_{IF}, and the spherical

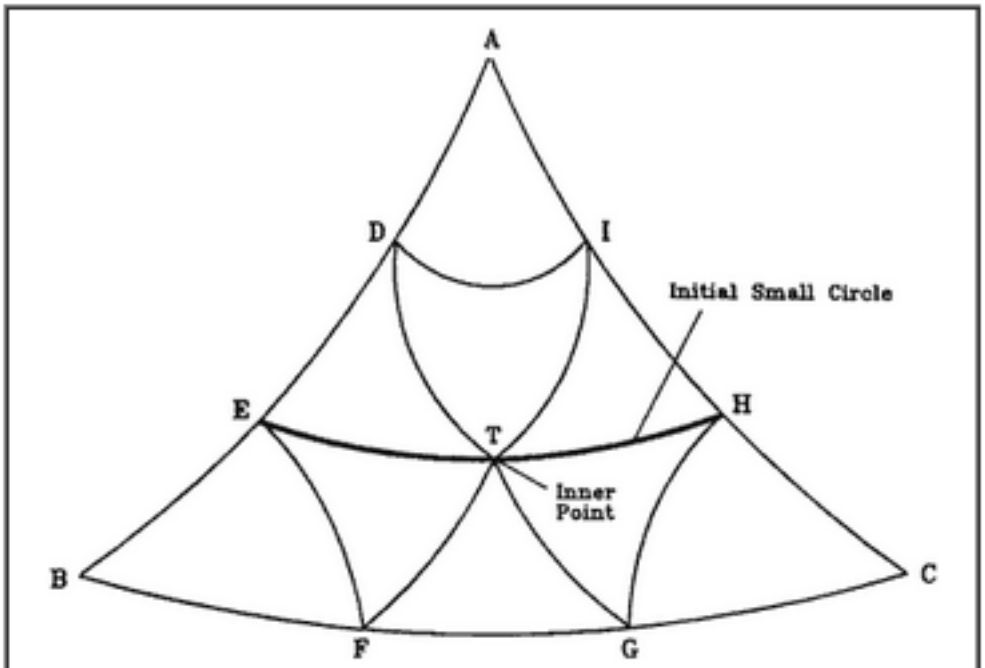


Figure 15. Nine-fold small circle subdivision with the inner point being the midpoint of the small circle EH which divides the small circle hexagon DEFGHI into two equal-area quadrilaterals.

coordinates of its pole P_{IF} as $(\theta_{P_{IF}}, \phi_{P_{IF}})$. The Cartesian coordinates of the poles P_{DG} and P_{IF} $(x_{P_{DG}}, y_{P_{DG}}, z_{P_{DG}})$ and $(x_{P_{IF}}, y_{P_{IF}}, z_{P_{IF}})$, can be calculated using **equation (1)**.

(2)

The angle between the vector (x_T, y_T, z_T) and the vector $(x_{P_{DG}}, y_{P_{DG}}, z_{P_{DG}})$ is a_{DG} and the angle between vector (x_T, y_T, z_T) and $(x_{P_{IF}}, y_{P_{IF}}, z_{P_{IF}})$ is a_{IF}. So the inner product gives:

(38)

$$x_T x_{P_{DG}} + y_T y_{P_{DG}} + z_T z_{P_{DG}} = \cos(\alpha_{DG})$$

(39)

$$x_T x_{P_F} + y_T y_{P_F} + z_T z_{P_F} = \cos(\alpha_{IF})$$

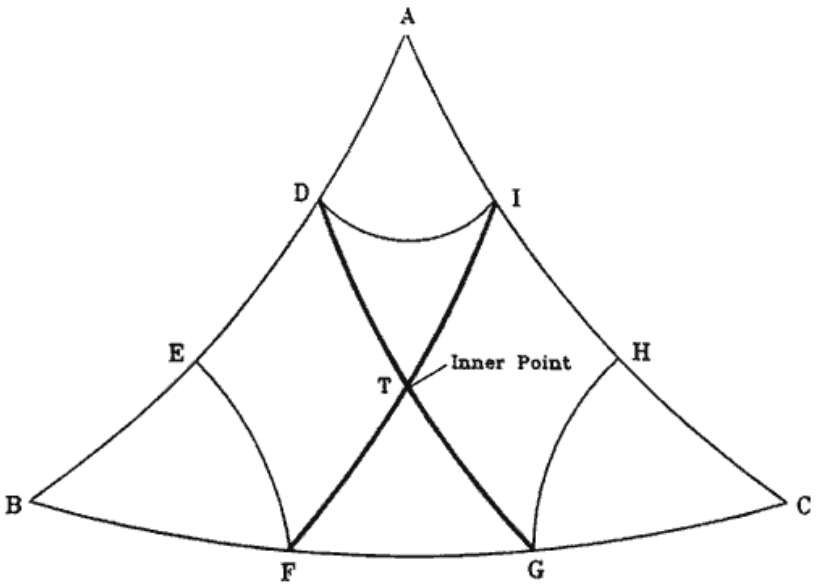


Figure 16. Inner point determination for nine-fold small circle subdivision. Each of the small circles DG and FI divides the hexagon DEFGHI into two equal-area quadrilaterals.

The intersected point T is on the unit sphere, so we get

(40)

$$x_T^2 + y_T^2 + z_T^2 = 1$$

Solving **equations (38)**, **(39)** and **(40)**, we get two solutions for the Cartesian coordinates of the intersection point T. We selected the solution close to vertex A.

After the trisection points and the inner point are found, we finally determine the six inner small circles DT, ET, FT, GT, HT and IT. There are six variables for the six relative co-latitudes of these small circles. If we apply the same method as in the four-fold subdivision case to the six inner sub-triangles so that each of them has an area one-ninth of the triangle ABC, six equations can be obtained. But only five of the equations are independent, so one more equation is needed to solve the six relative co-latitudes. The sixth equation can be obtained by maximizing the average compactness of the six sub-triangles. Although this method is ideal and

guarantees the equal area and maximum compactness, it is practically impossible to implement because the six equations are nonlinear. Instead, we assume at least one initial small circle to be known a priori. The choice of the initial small circles depends on the determination of the inner point. In**Figure 15**, if we select the midpoint T of the small circle segment EH as the inner point, we can assume the segments ET and HT to be two initial small circles and their relative co-latitudes are the same as the relative co-latitude of the small circle EH. Thus, the relative co-latitudes of other four small circles can be easily calculated one-by-one using the method introduced previously for four-fold subdivision. In **Figure 16**, if we select the intersection point T of the two small circles DG and IF as the inner point, one of the small circle segments DT, GT, IT and FT can be selected to be an initial small circle. The relative co-latitude of the initial small circle is the same as that of the small circle DG or IF. In the figure, we have selected the segment FT as an initial small circle. The relative co-latitudes of other five small circles can be derived one-by-one using the same method as in the four-fold subdivision case.

After the relative co-latitudes of the nine small circles are computed, the poles and the arc lengths of all the small circles can be calculated the same way as in the four-fold case.

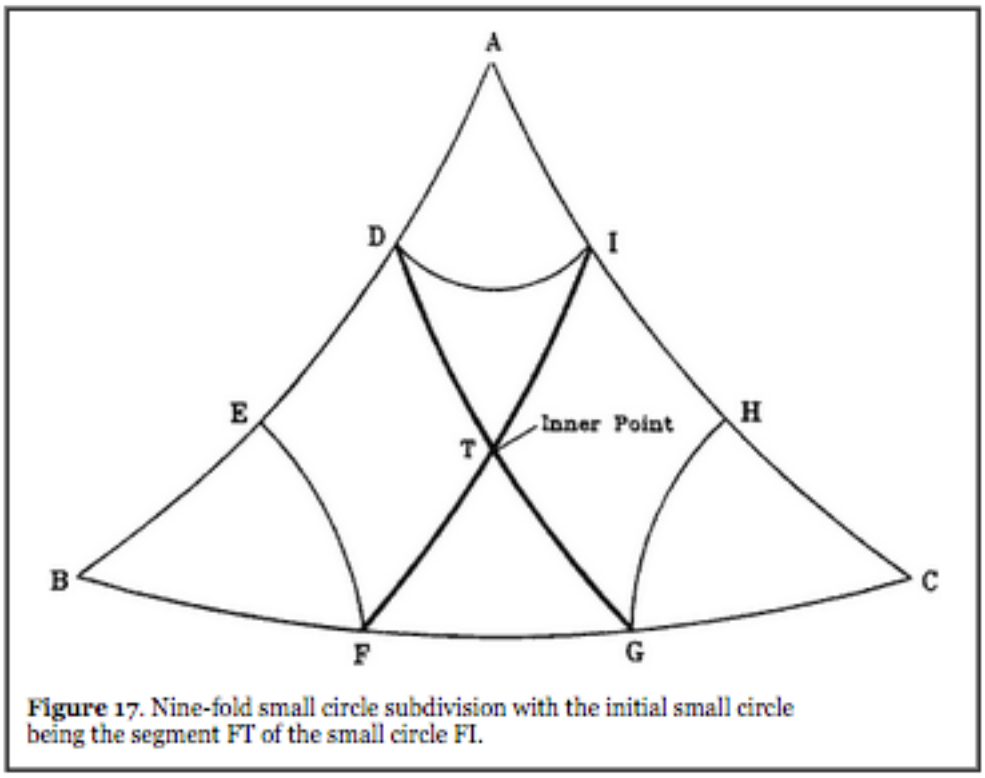


Figure 17. Nine-fold small circle subdivision with the initial small circle being the segment FT of the small circle FI.

SMALL CIRCLE METHOD EVALUATION

The small circle subdivision method can be evaluated in many ways, such as to what degree it satisfies the Goodchild criteria for global grids (Kimerling et al. 1999). We have chosen, however, to focus on two primary Goodchild criteria, area variation and high compactness constancy. Area and compactness metrics were computed for several competing methods, but we focus on comparisons with the ISEA grid which has received serious consideration due to its equal area nature and compactness consistency.

Spherical Area Variation

The area computations central to small circle partitioning should insure that the sub-triangles will be identical in spherical surface area at all levels of recursion. To prove that the equation coding was correct, computed triangle areas were stored up to 9-fold partitioning, recursion

level 8 (43,046,721 triangles, each 0.59 km^2 on the globe). Statistical analysis of these data showed a range, and hence standard deviation, of 0.0 for each level of recursion, the same result obtained for ISEA triangles. For each level of recursion, there is no variation in triangle areas on the sphere.

Compactness Variation

The compactness of grid cells is an important geometrical property, as the ideal global grid would have identically shaped cells (zero compactness variation) at the highest possible compactness. The first way to study compactness variation is through visual examination of grid cell appearance across the icosahedron face. Comparing the 4-fold subdivision, recursion level 4 triangular grid cells for the ISEA method and from the small circle subdivision method (**Figure 18**), the smoother, more regular cell edges resulting from small circle subdivision are

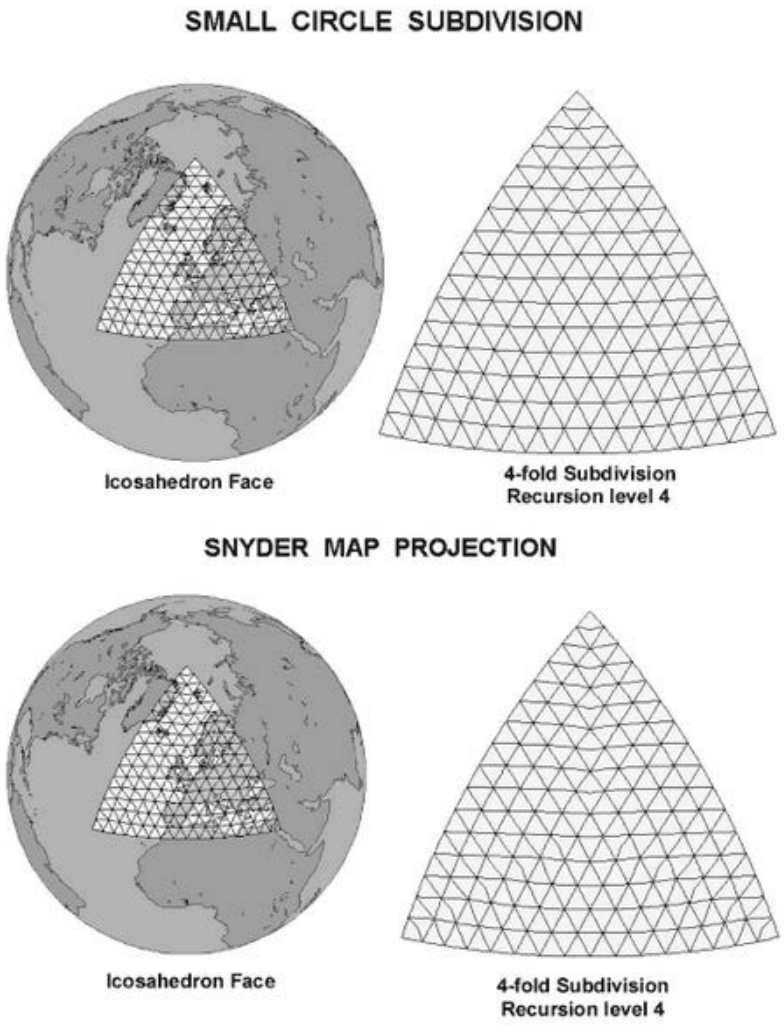


Figure 18. Small circle and Snyder (ISEA) subdivision of a spherical icosahedron face using 4-fold subdivision at recursion level 4. The icosahedron face is shown on an orthographic map projection to show the earth's sphericity.

readily apparent. Cell edge variations in the ISEA grid appear to be localized along three lines radiating from the center of the icosahedron face to each corner, bisecting the vertex angles. Triangle edges along these lines appear to alternately bow inward and outward relative to geodesic lines connecting two corners. Triangle edges away from these three lines quickly smoothen and triangles appear more equilateral in shape and similar in compactness.

Visual observation of shape and compactness variation can be augmented by quantitative analysis of compactness variation among cells if a suitable compactness metric exists. The Zone Standardized Compactness (ZSC) measure, described in Kimerling et al. (1999) was used so that

comparisons could be drawn with ZSC values for ISEA grid triangles at different recursion levels. ZSC values are scaled relative to a 1.0 value for a spherical zone (cap) of the same surface area as the sub-triangle.

ZSC values for 4-fold small circle subdivision at recursion levels 1-6 were computed, with the values for recursion levels 2,4 and 6 plotted on the equilateral triangle diagrams (not the actual sub-triangle shapes) in Figure 19, along with corresponding ZSC values for the ISEA method. The spatial pattern of ZSC compactness variation for small circle subdivision at first appears chaotic, but subtle symmetries along three lines bisecting the vertex angles soon become apparent. At all levels of recursion the three corner triangles, along with several interior triangles along the bisector lines, have the highest compactness. Note that the minimum and maximum compactness values seen on the diagrams and in the legend define the range of compactness for the ISEA grid, but for both methods compactness values are well below and above the 0.7776 compactness of a spherical equilateral triangle. This is because for both methods the triangle edges along the bisector lines bow outward and inward, alternately increasing and decreasing triangle compactness. However, the amount of bowing, and hence the compactness range, is much greater with the ISEA method. The triangular diagrams for small circle subdivision recursion levels 4 and 6 show that this

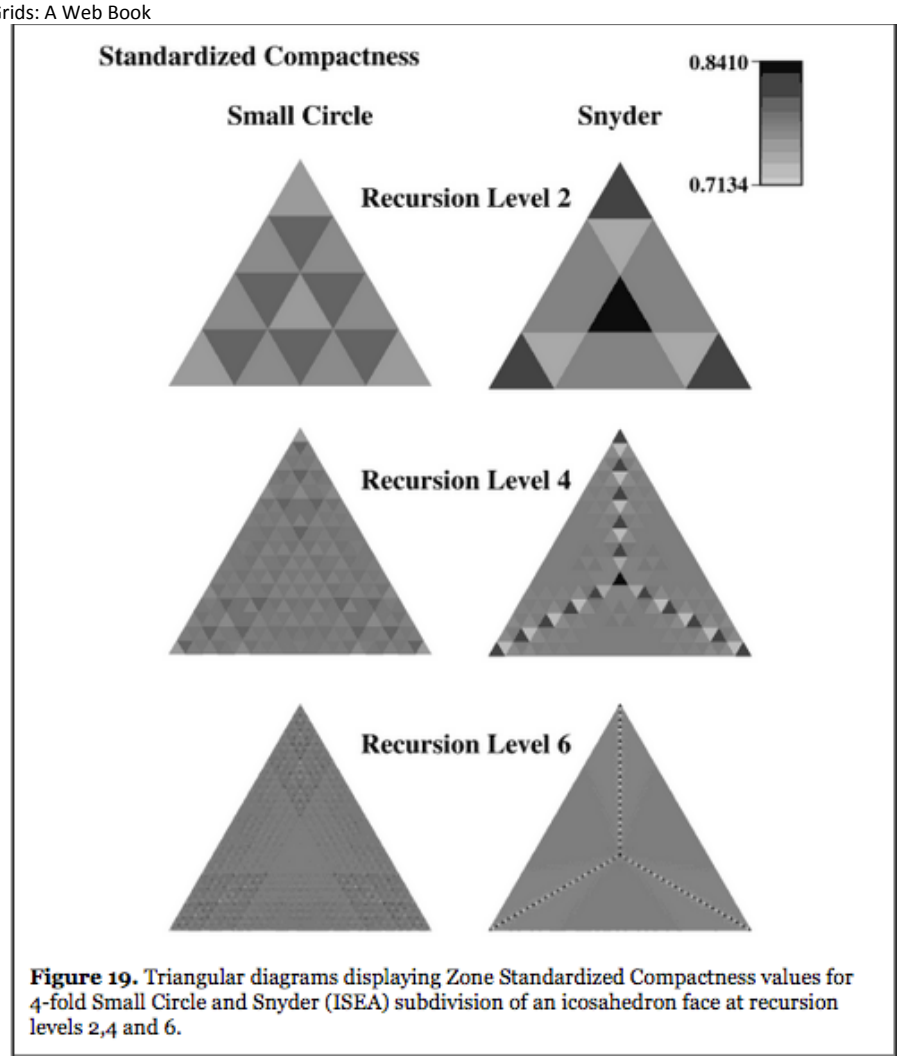


Figure 19. Triangular diagrams displaying Zone Standardized Compactness values for 4-fold Small Circle and Snyder (ISEA) subdivision of an icosahedron face at recursion levels 2,4 and 6.

alternating inward and outward bowing of small circle edges to achieve equal area sub-triangles continues at higher recursion levels. The general pattern is that the higher compactness sub-triangles at the next lower recursion level are partitioned similarly, with center triangles of lower compactness surrounded by three higher compactness outer triangles. However, the pattern is reversed for the center triangles of lower compactness. Here a center triangle of higher compactness is created, surrounded by three lower compactness outer sub-triangles. This roughly fractal-like "triangular checkerboard" pattern of sub-triangle compactness continues at higher recursion levels, but the compactness differences among adjacent sub-

triangles lessen except at the initial icosahedron corners where both the highest and lowest compactness sub-triangles are always found.

Statistical analysis of compactness in spherical triangle grid cells gives us additional ways to compare small circle subdivision with other partitioning methods such as the ISEA. The range in compactness for the small circle method (**Figure 20**) shows a slight increase for 4-fold subdivision with increasing levels of recursion because the inward or outward bowing of sub-triangle edges becomes greater at higher recursion levels. However, these ranges are less than those for the ISEA method.

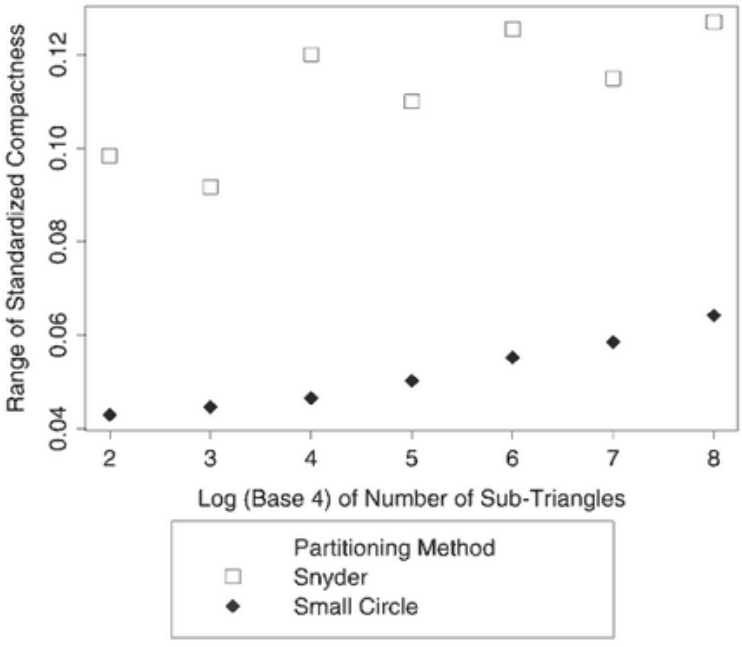


Figure 20. Range of zone standardized sub-triangle compactness for the Small Circle and Snyder (ISEA) methods for different 4-fold subdivision levels of recursion.

The mean ZSC at different recursion levels behaves similarly for both 4- and 9-fold subdivision, with 4-fold subdivision results shown in (**Figure 21**). There is with a rapid decrease in ZSC to a stable minimum value at higher recursion levels. The small circle means are higher than those for the ISEA method, but less than the means for equilateral triangles. Similar to the ZSC mean values, the standard deviations show a decrease for both 4- and 9-fold subdivision to a stable

minimum at higher recursion levels (**Figure 22**). The minimum standard deviation reached for small circle subdivision is lower than that for the ISEA method.

These results for the full population of triangular cells at recursion levels 2-8 show the small circle method to perform better both in terms of the average compactness produced and the variation in compactness among sub-triangles.

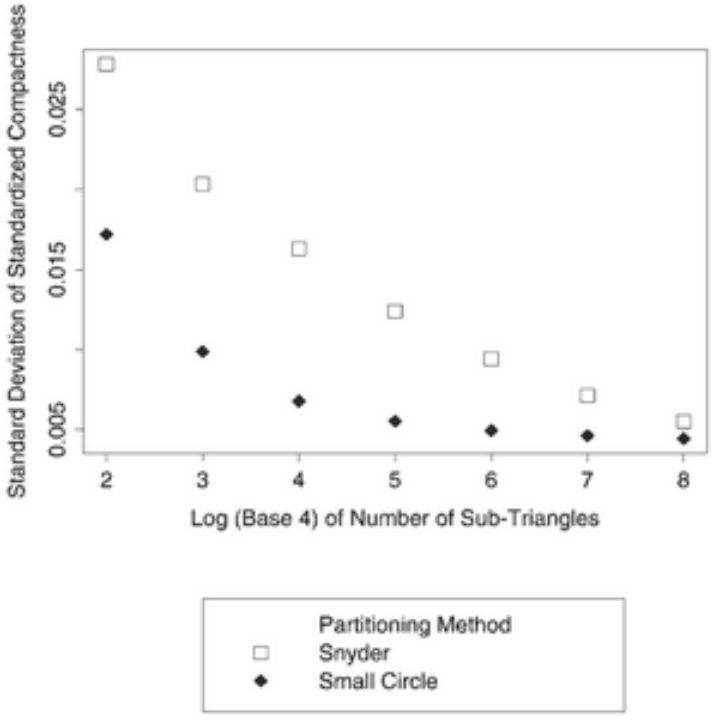


Figure 21. Standard deviation of zone standardized sub-triangle compactness for the Small Circle and Snyder (ISEA) methods for different 4-fold subdivision levels of recursion.

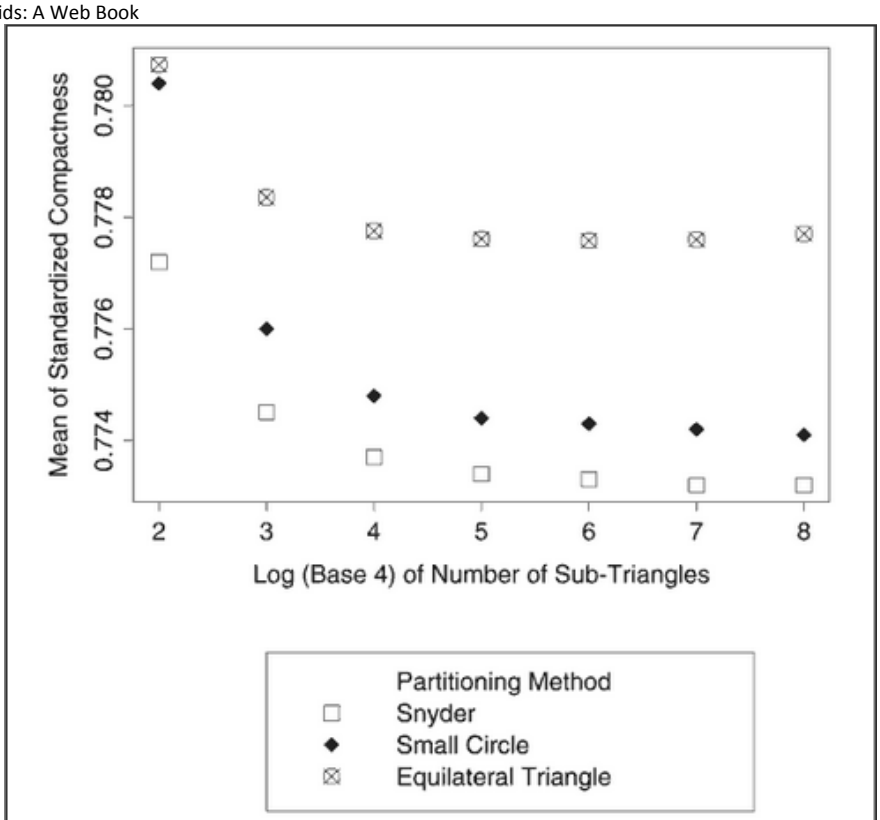


Figure 22. Mean of zone standardized sub-triangle compactness for the Small Circle and Snyder (ISEA) methods for different 4-fold subdivision levels of recursion.

CONCLUDING REMARKS

We have conducted a complete analysis of the small circle subdivision method, including its mathematical derivation, numerical implementation and detailed analysis of results. We have also performed a comprehensive comparison of the method with five other global partitioning methods, although only the ISEA compactness results are presented here. The numerical comparison methods discussed here are closely linked to survey sampling requirements for a global grid, as expressed in the Goodchild criteria (Kimerling et al. 1999). We focused on the comparison of surface area and compactness variations when identical subdivision frequencies and recursion levels were applied to these partitioning methods.

The comparison results show that although the small circle subdivision, the ISEA, and the Tobler-Chen methods satisfy the equal-area grid cell criterion, the small circle subdivision method creates at each level of recursion triangular cells with a higher mean compactness,

lower compactness range and smaller standard deviation in compactness. Thus, if the equal-area sampling units are a prime requirement, the small circle subdivision method is the best option among the partitioning methods examined here.

Although the small circle subdivision method has better compactness characteristics than the ISEA or Tobler-Chen method, its mean compactness is lower and the range and standard deviation in compactness are large when compared to other non equal-area partitioning methods not examined in this article, such as the Fuller-Gray and Gnomonic map projection surfaces (White et al. 1998). This compactness variation is more obvious at higher 9-fold subdivision recursion levels. One of the factors contributing to the compactness variation is the location of the inner point in 9-fold subdivision. If we locate the inner point in 9-fold subdivision by maximizing the mean compactness of the surrounding six sub-triangles, the compactness characteristics could be improved, but it is computationally complex.

Computational complexity, not geometrical regularity, appears to be the major drawback to widespread use of the small circle subdivision method. C-language code for its implementation, available from the authors, bears this out. However, run times for higher recursion levels were not prohibitively longer than the ISEA method, so perhaps the ever increasing CPU power will soon make the small circle method computationally practical as well as geometrically superior to other global grid alternatives. Computation speed may also be increased significantly by converting the equations based on spherical trigonometry into their vector algebra equivalents. This would very likely result in a simpler code used to generate the numerical solution.

REFERENCES

Baumgardner, J.R. and P.O. Frederickson. 1985. Icosahedral Discretization of the Two-Sphere. *S.I.A.M. Journal of Numerical Analysis*. 22(6): 1107-15.

Brooks, D.R. 1981. Grid Systems for Earth Radiation Budget Experiment Applications.*NASA Technical Memorandum* 83233. 40 pp.

Coexeter, H.S.M. 1962. The Problem of Packing a Number of Equal Nonoverlapping Circles on a Sphere. *Transactions, New York Academy of Science, Division of Mathematics*, Series II, vol. 24: 320-31.

Dutton, G.H. 1999. *A Hierarchical Coordinate System for Geoprocessing and Cartography*. Berlin: Springer-Verlag. 230 pp.

Goodchild, M.F. 1994. Geographical Grid Models for Environmental Monitoring and Analysis across the Globe (panel session). GIS/LIS '94 Conference, Phoenix, AZ.

Kahn, R. and A. Braverman 1999. What Shall We Do with the Data we are Expecting From Upcoming Earth Observation Satellites? *Journal of Computational and Graphical Statistics*, 8(3): 575-588.

Kimerling, A.J., Sahr, K., White, D. and L. Song. 1999. Comparing Geometrical Properties of Global Grids. *Cartography and Geographic Information Science*. 26(4): 271-287.

Rakhmanov, E.A., Saff, E.B., and Y.M. Zhou. 1994. Minimal Discrete Energy on the Sphere. *Mathematical Research Letters*. 1:647-62.

Saff, E.B. and A.B.J. Kuijlaars. 1997. Distributing Many Points on a Sphere. *The Mathematical Intelligencer*. 19(1): 5-11.

Snyder, J.P. 1992. An Equal-Area Map Projection for Polyhedral Globes. *Cartographica*. 29(1):10-21.

Tobler, W.R. and Z. Chen. 1986. A Quadtree for Global Information Storage.*Geographical Analysis*. 18(4): 360-71.

White, D., Kimerling, A.J., Sahr, K. and L. Song. 1998. Comparing Area and Shape Distortion on Polyhedral-based Recursive Partitions of the Sphere. *International Journal of Geographical Information Science*. 12: 805-27.

Chapter Two:

Interoperable Coordinate Transformation and Identification of Coordinate Systems

By Daniel Specht

Engineer Research and Development Center

Specht, D. (2002). Interoperable Coordinate Transformation and Identification of Coordinate Systems. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis.

ABSTRACT. The OpenGIS Consortium (OGC), an industry group, has developed a Java interface specification allowing vendors to develop mutually interoperable coordinate transformation components for geospatial software. OGC established the Coordinate Transformation Working Group (CT-WG) in April 1998. The working group developed a Unified Modeling Language object model of coordinate systems and coordinate transformations in coordination with ISO/TC211; OGC accepted interface specifications drafted using this model. The CT-WG also released an XML DTDdescribing coordinate systems and coordinate transformations.

This paper describes the object model, the XML and the specifications. The object model describes a coordinate system as a collection of axes and at least one datum.

The specification provides a common way of identifying coordinate systems and of accessing coordinate transformation services that support accuracy calculation. When implemented, these specifications will ease data import; users of compliant applications will import data unaware of its coordinate system. If the application cannot import data in a given coordinate system, a compliant server will transform the coordinates to the native coordinate system.

Introduction

The OpenGIS Consortium (OGC) is a non-profit organization that promotes interoperable geoprocessing. The goal is that geospatial applications are interoperable. This means they search for, access and display data and services using standard interfaces. Clients using standard interfaces need not know much about the server. OGC defines OpenGIS as "transparent access to heterogeneous geodata and geoprocessing resources in a networked environment." OGC's goal is open interface specifications enabling developers to write interoperating components providingOpenGIS."

Representatives of software vendors, government agencies and academic institutions that belong to OGC write requirements for interface specifications. These requirements are published as a Request for Proposal (RFP). Member organizations then submit proposals for these specifications. After OGC accepts a proposal as an Implementation Specification OGC submits it to the International Standards Organization Technical Committee 211, ISO/TC211 for acceptance as an ISO standard.

Interface standards are widely used outside of the geospatial community. For example, in the Microsoft environment a user can cut out a piece of a spreadsheet and drop it into any conformant application (e.g., MS Word). The application will have access to spreadsheet services and data stored on the spreadsheet. To the user it appears as if the spreadsheet is running on the application. What we cannot do yet is to tell our spreadsheet that column H is Transverse Mercator coordinates and please change them to Lambert coordinates using a coordinate transformation service on the network.

However OGC has developed several sets of geospatial interfaces. These support interoperable commercial web GIS applications (listed at http://www.opengis.org/cgi-bin/implement.pl).

The Abstract Model of Coordinate Transformation

The US Army struggled with poor coordinate transformation software for years. Interoperable coordinate transformation software would help solve this problem. With this in mind I chartered a coordinate transformation Working Group (CT-WG) at OGC. The group developed the Abstract Model, an object model of coordinate transformations and coordinate systems using Unified Modeling Language (UML), a notational language used for object oriented analysis and design. UML is widely used and has been standardized by the Object Modeling Group.

Figure 1 shows what a UML object class looks like.

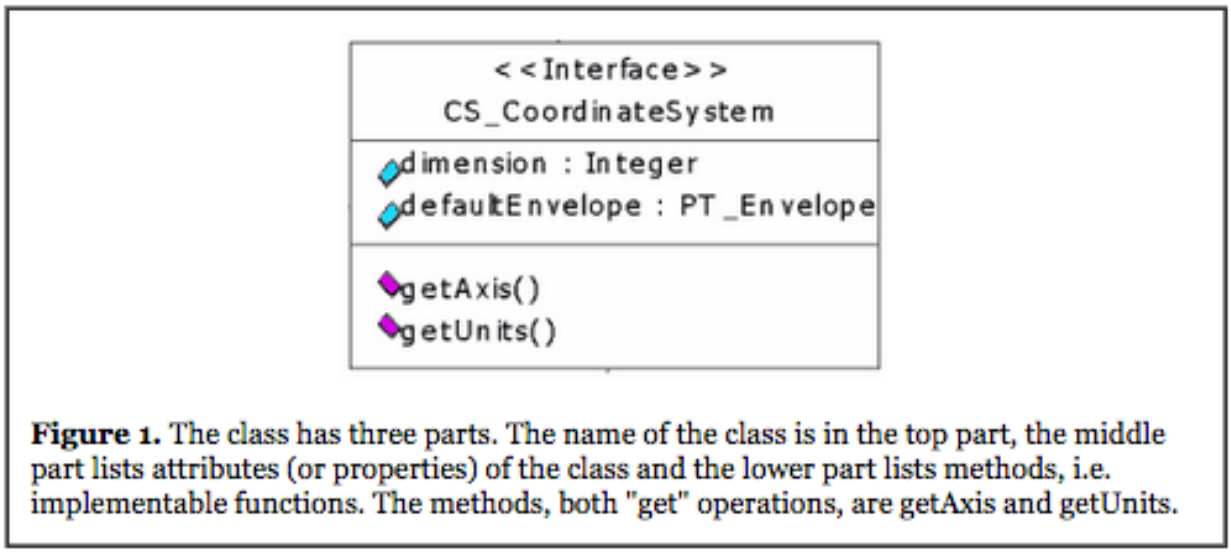


Figure 1. The class has three parts. The name of the class is in the top part, the middle part lists attributes (or properties) of the class and the lower part lists methods, i.e. implementable functions. The methods, both "get" operations, are getAxis and getUnits.

XML

Because more applications are using Extensible Markup Language (XML), the CT-WG wrote an XML DTD to describe coordinate systems and coordinate transformations (Whiteside, 1999). The DTD does not contain all the information in the UML model, because XML doesn't support inheritance and aggregation.

Achieving Consensus

The CT-WG struggled to reach a consensus on what the model should look like while coordinating with our counterparts at ISO/TC211 to maintain conformance between the ISO/TC211 model and our own. This was not always easy.

Consider a point stored as latitude, longitude and elevation above sea level. Latitude and longitude are referenced to a geodetic datum while elevation is referenced to a vertical datum.

ISO/TC211 calls this a compound coordinate reference system, composed of two coordinate reference systems, each with its own datum, shown in **Figure 2**. (The object at the pointy end of the triangle is the superclass and the object at the other end is the subclass.)

Some OGC members felt a compound Coordinate Reference System (CRS) should be expressed as a single CRS with two datums and that the model should not include a compound CRS (**Figure 3**). There were other proposals as well.

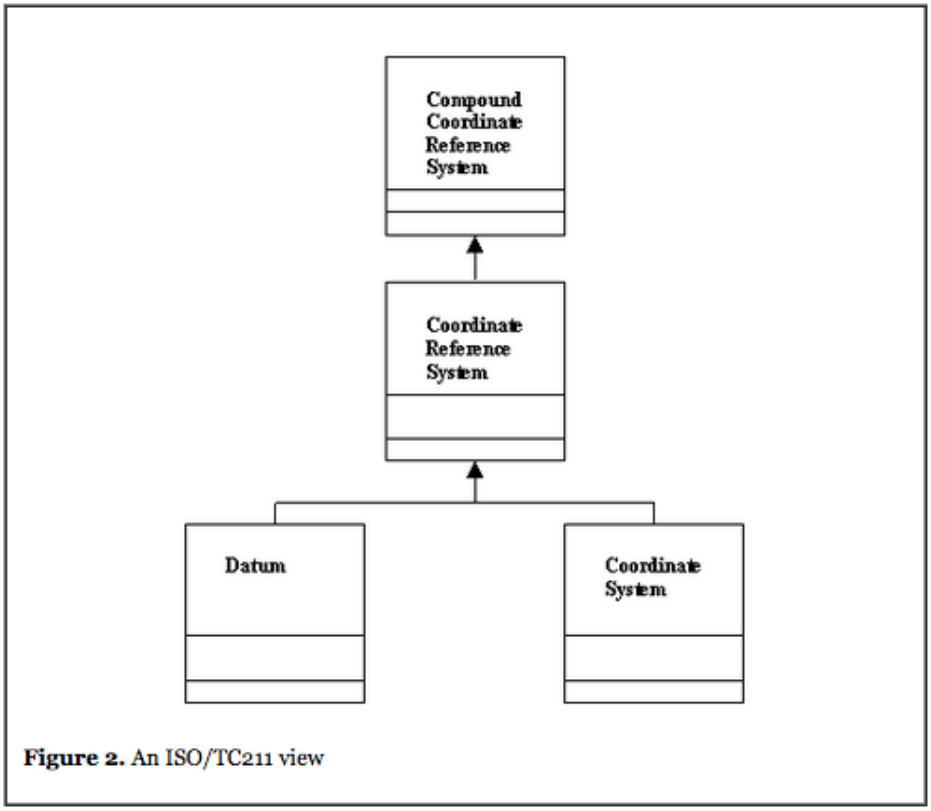


Figure 2. An ISO/TC211 view

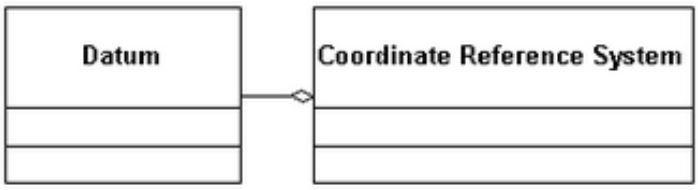


Figure 3. An early OGC view; the diamonds on a stick indicate strong aggregation, i.e., the datum is part of the coordinate Reference System.

The CT-WG could not come to a consensus and created a UML model that left the issue open. This UML model is part of the OpenGIS Abstract Specification that was included in the RFP.

OGC asked members to propose an Implementation Specification that solves this problem. The proposed solution is shown in **Figures 4** and **5**. The revised UML model was published as part of the implementation specification.

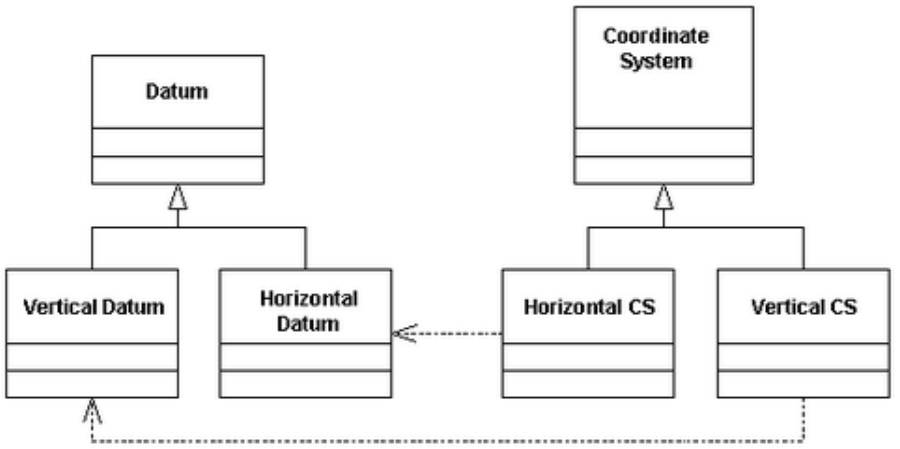


Figure 4. A representation of the model used in the Implementation Specification. The dotted line with the arrows indicates dependency; if the datum goes away so does the coordinate system. These relationships are not explicit in the actual UML model, part of which is shown (in a simplified form) in **Figure 5**.

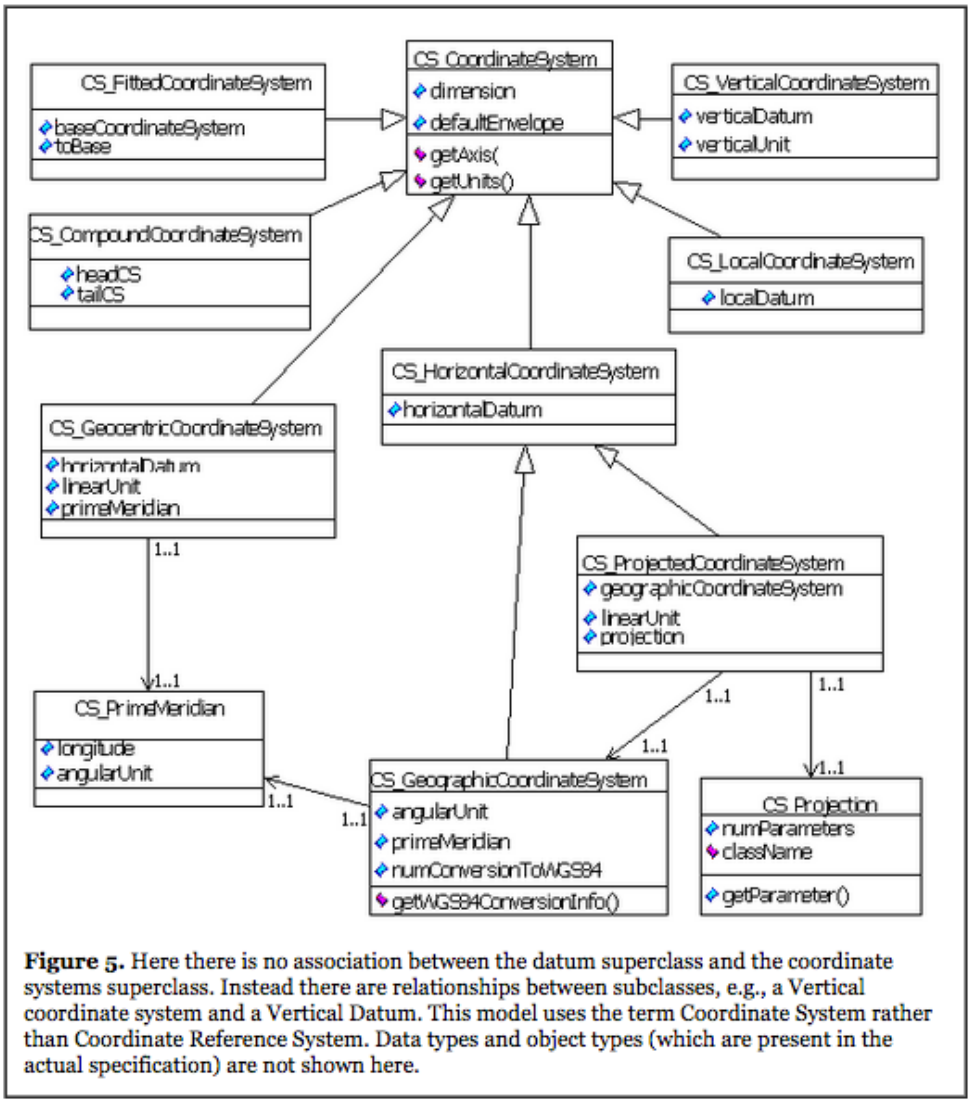


Figure 5. Here there is no association between the datum superclass and the coordinate systems superclass. Instead there are relationships between subclasses, e.g., a Vertical coordinate system and a Vertical Datum. This model uses the term Coordinate System rather than Coordinate Reference System. Data types and object types (which are present in the actual specification) are not shown here.

Implementation Model

The object model for coordinate systems includes a number of coordinate system classes and a number of datum classes.

A **Compound Coordinate System** includes a horizontal and a vertical coordinate systems and references two datums, a horizontal (geodetic) and a vertical datum. However where the Compound Coordinate System is 3-D ellipsoidal both the horizontal and vertical components may reference the same datum.

A **Horizontal Coordinate System** is a 2-D coordinate system that may be Projected (a cartographic projection) or Geographic (latitude and longitude). Horizontal coordinate systems use horizontal (geodetic) datums.

A **Vertical Coordinate System** is a vertical 1-D coordinate system (e.g., vertical with respect to a geoid or ellipsoid). This references a vertical datum such as a sea level datum or a horizontal (geodetic) datum.

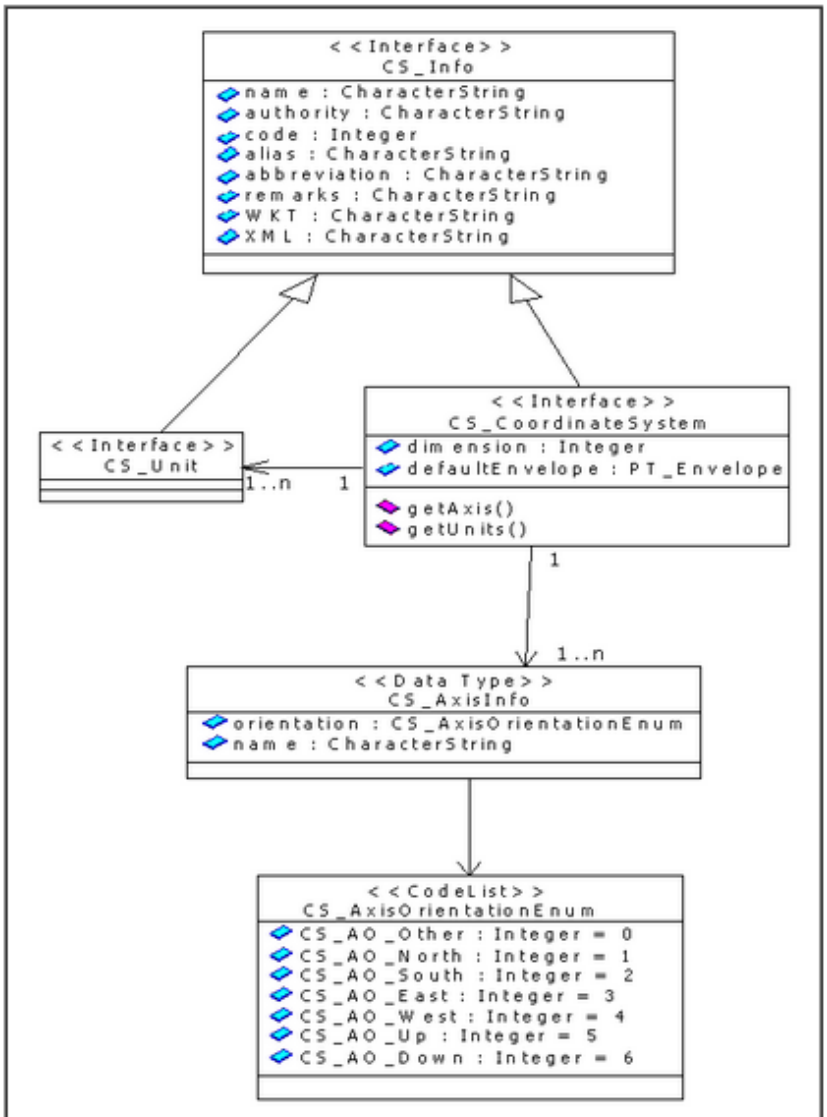


Figure 6. This specification supports users who create a coordinate system in any number of dimensions.

A **Local Coordinate System** references a local datum. A Local Coordinate System cannot be transformed to another coordinate system. Once the Local Coordinate System is georeferenced it is no longer a Local Coordinate System. Note that a local datum is not necessarily 2-D Cartesian.

Geocentric Coordinate Systems are 3-D Cartesian geocentric Coordinate Systems that reference a horizontal datum.

Fitted Coordinate Systems sit "inside another coordinate system. The fitted coordinate system can be rotated and shifted, or use any other math transform [a mathematical function] to inject itself into the base coordinate system" (OpenGIS Implementation Specification, 2001). A Fitted Coordinate System references the datum of the base Coordinate System.

This specification will also support creation of coordinate systems of any dimension, because each of these coordinate systems and datums have the metadata shown in**Figure 6**, representing ellipsoid, geoid, prime meridian, axis, parameters (both projection and transformation parameters) and unit (i.e. linear or angular units). The specification also includes enumerated lists for things like datum type (e.g., Altitude Barometric and Orthometric) and axis orientation. Attributes in CS_Info are from ISO/TC211 Spatial Referencing by Coordinates.

Identification

Interoperability requires a universal way to identify a coordinate system, even if the coordinate system is unique to a small project. A client cannot import data without knowing the coordinate system. However, no one has volunteered to keep the list of all coordinate systems for the entire geospatial community. The best anyone can do is to keep a list of lists.

Our coordinate system identifier will consist of an authority (one of the lists within the list), an edition (because an authority may update its list) and a code (a string). The authority is whoever assigned the string to the coordinate system. This system is distributed, extensible and supports legacy name lists (if they are translated into integers).

This will only work if there is a single authority. However a Compound Coordinate System may have a vertical coordinate system from one authority and a horizontal coordinate system from another.

Interfaces

The specification provides interfaces for coordinate transformation (not discussed in this paper), to create a coordinate system and to access coordinate system metadata (i.e., attributes or properties). The interfaces support accuracy of a transformation and of a coordinate, but not precision of a coordinate system.

The specification originally included three profiles (i.e., versions) with analogous functionality.

1) Microsoft Interface Definition Language (MIDL) specifications for COM interfaces
2) Interface Definition Language (IDL) specifications for CORBA interfaces
3) Java source specifications for Java interfaces.

OGC only released the Java interfaces.

Because most of the recent work at OGC uses HTTP, there may be an HTTP profile at some point.

CONCLUSIONS

When widely implemented, the coordinate transformation specification will provide a common way of identifying coordinate systems and of accessing coordinate transformation services that support accuracy calculation. These specifications will ease data import. Users of compliant applications will import data without having to be aware of the coordinate system of the imported data because the client and server will use the same identifier for a given coordinate system. If the application cannot transform imported data to the native coordinate system, a compliant server will transform the coordinates to the native coordinate system. Compliant

software will therefore import data more reliably and be able to deal more effectively with distributed data.

When I presented this paper in March 2000 a GIS vendor (Cadcorp Ltd, UK) had already fully implemented the COM profile of the coordinate transformation specification. The specification supports the software's ability to transform an image or a feature (i.e. a feature geometry). **Figure 7** shows how this transforms an image.

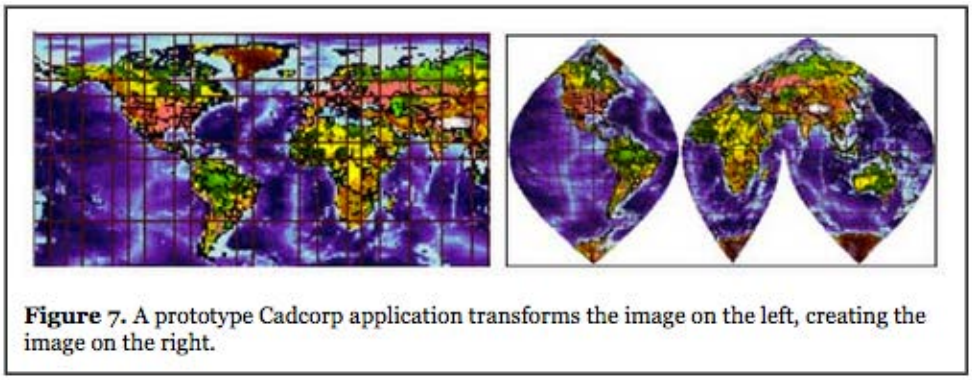


Figure 7. A prototype Cadcorp application transforms the image on the left, creating the image on the right.

The image on the left in **Figure 7** was downloaded using the Open GIS Web Map Server Interface Specification. Web browsers can as of January 2002 overlay geospatial data views from servers made by 22 software producers using this specification. Overlay will be easier using interoperable coordinate transformation software.

As of January 2002 there have been no additional implementations of the Coordinate Transformation specification. OGC-compliant applications identify coordinate systems using EPSG codes. Although the number of OGC-compliant commercial products is growing rapidly, new products largely use HTTP interfaces. Commercialization of the coordinate transformation interface will probably be limited until the release of an HTTP interface.

OGC maintains the coordinate transformation specifications, improving them based on feedback from software developers and developing additional coordinate transformation interface specifications.

REFERENCES

2001, OpenGIS Implementation Specification: Coordinate Transformation Services Revision 1.0, OpenGIS Project Document 01-009, Cadcorp Ltd.http://www.opengis.org/techno/specs/01-009.zip

2000, Geographic information - Spatial referencing by coordinates, Draft International Standard 19111, ISO/TC211 N934, http://www.opengis.org/techno/specs/01-009.pdf

1999, OGC Request 9: Core Task Force, Coordinate Transformation Working Group, A Request for Proposals: OpenGIS Coordinate Transformation Services, OpenGIS Consortium

1999, OpenGIS Abstract Specification, Topic 2: Spatial Reference Systems, Version 4, Project Document 99-102r, http://www.opengis.org/techno/abstract/99-102r1.pdf

de la Beaujardiere, J., 2002, OpenGIS Implementation Specification: Web Map Server Interfaces Implementation Specification, Version 1.1.1, OpenGIS Project Document 01-68r3, http://www.opengis.org/techno/implementation.htm

Fowler, M. and Scott, K., 1997, UML Distilled, Applying the Standard Object Modeling Language, Addison Wesley

Whiteside, A. and Nicolai, R. 1999 , Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations, Version 1.1.0,http://www.opengis.org/techno/discussions/01-014r5.doc

ACRONYMS AND ABBREVATIONS

COM
: Common Object Model

CORBA
: Common Object Request Broker Architecture

CRS
: Coordinate Reference System

CT-WG
: Coordinate Transformation Working Group

DTD
: Data Type Definition

ISO/TC211
: International Standards Organization Technical Committee 211

OGC
: OpenGIS Consortium

RFP

Request For Proposal

UML

Unified Modeling Language

XML

eXtensible Markup Language

Chapter Three:

Discovering, Modeling, and Visualizing Global Grids over the Internet

By Yvan G. Leclerc, Martin Reddy, Lee Iverson and Michael Eriksen

SRI International, Menlo Park, CA, USA

Leclerc, Y. G., Reddy, M., Iverson, L., & Eriksen, M. (2002). Discovering, Modeling, and Visualizing Global Grids over the Internet. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis.

Abstract. Three integrated tools are presented to solve the problems of: 1) discovering appropriate global grid and associated data on the Internet, 2) modeling complex 3D structures using a standards-based geospatial reference model, and 3) visualizing global grids using 3D Web-streaming technology. Specifically, the GeoWeb is an infrastructure for indexing and discovering any data on the Internet based upon its location, GeoVRML is an open standards-based file format for georeferenced 3D models that is undergoing ISO standardization, and TerraVision is a freely-available Web-based 3D terrain visualization system. These components are integrated in the TerraVision client browser, which can automatically query the GeoWeb to discover all available data for the region being viewed and which can render GeoVRML format data overlaid on a global grid that is streamed incrementally over the Web using view-dependent techniques.

Introduction

This chapter presents a collection of integrated tools that address the problems of discovering appropriate global grid data on the Internet, modeling associated 3D data in a standards-based file format that supports accurate georeferencing, and producing compelling 3D visualizations of all these where data can be progressively streamed over the Internet. We begin by describing the GeoWeb infrastructure, a distributed geographic index for the Web built on top

of the Domain Name System (DNS). We then describe the GeoVRML file format, an open extension to the Virtual Reality Modeling Language (VRML) that supports geographic coordinate systems and fusion of disparate data sources. Finally, we describe our TerraVisionTM terrain visualization system, a cross-platform application for browsing massive multiresolution terrain databases over the Web in 3D.

Taken as a coherent suite of solutions, these products provide the ability to index all geographic data on the Web in an open and highly scalable infrastructure, to support the use of accurately georeferenced 3D models using an ISO standard data format, and to enable users around the world to discover and visualize all these data using freely-available 3D clients for their desktop computers.

The GeoWeb Infrastructure

The Web has revolutionized the way documents are published, found, and viewed. This revolution has succeeded in part because Web protocols and publishing standards are open, and there exist freely available tools for publishing and reading Web documents. Just as importantly, powerful search engines let users find documents of interest almost instantaneously. Without these search engines, the Web would be almost useless.

Even though search engines can now find almost all Web-accessible documents about a given topic, it is currently difficult to find Web-accessible data about a given location because the location is typically not a part of the data themselves. Examples of such georeferenced data include aerial and satellite images, 3D models of buildings and weather systems, and vacation pictures taken with GPS-enabled cameras. Thus, today's search engine technology is not applicable and the huge amount of georeferenced digital data that are available on the Web today is virtually inaccessible.

In an attempt to make georeferenced data accessible, a number of companies and organizations have created private or governmental databases that hold a small fraction of all

georeferenced data. For example, companies like MapQuest maintain private map databases with associated street addresses and links to businesses like restaurants and shops. These databases hold searchable metadata, a summary description of the data that includes geographic location, which can either be searched directly, or via a clearinghouse. However, none of these organizations is either capable of, or willing to, create a database that would allow anybody in the world to publish and search georeferenced metadata. Indeed, this task is so large that no single, regional, organization could do it alone.

What is needed instead is a coordinated global infrastructure with participating organizations from around the world. We call this open standards-based infrastructure the GeoWeb, which could be realized as a new top-level domain (e.g., .geo) that would enable anybody to publish and search for all metadata referring to a given area (Leclerc et al., 2001a; Leclerc et al., 2001b; Leclerc et al., 2000). The infrastructure is based on a hierarchy of servers whose domain names represent geographic areas. An example hierarchy, described below, is nominally of the form minutes.degrees.tendegrees.geo. (For convenience, we use ".geo" as the top-level domain name in all of our examples of the GeoWeb hierarchy, although existing top-level domains could be used such as ".dotgeo.net".)

Using the GeoWeb hierarchy, Web sites or specialized applications will let users specify a search profile (keywords and other pertinent information) and find all georeferenced data satisfying the profile in a given area. Because the data are georeferenced they can be embedded in 2D maps or 3D models of the Earth, letting the users navigate through the map or model, providing a much more natural means of finding data than traditional search engines.

GeoWeb Design Principles

The GeoWeb is a distributed searchable database of metadata. It comprises the metadata standard, the distributed searchable database in the form of a DNS hierarchy of servers, and an API library for publishing, searching, and validating/endorsing metadata in the servers. The

GeoWeb is designed according to a few basic principles, similar in many ways to the Alexandria Digital Library (see http://www.alexandria.ucsb.edu/):

1. **Naive transparency**: Mapping from categories to metadata content is syntactically and semantically trivial. It should not be a difficult task to put together a sophisticated query involving a combination of criteria for a search.
2. **Latitude/longitude and temporal extent**: Since the basic search criteria are lat/long, elevation, and time, these must be explicit and required fields in the metadata.
3. **Efficient coverage**: In general, search fields should all be of relatively equal relevance. So that there is no overhead due to making something explicitly available for search that is rarely used.
4. **Abstract query formats**: One should explicitly avoid exposing any of the internal structure of the database in order to formulate queries. Moreover, one needs to allow for the possibility of sophisticated Boolean combination of simple queries in order to find exactly the desired set of metadata.
5. **Standard format**: It is desirable to adopt or at minimum support one of the existing standard metadata formats for external publishing (e.g., Dublin Core, FGDC, ISO/TC211).
6. **Multilingual**: It is important to respect the need to publish and discover metadata in languages other than English. This demands support for both multilingual descriptions and cross-linguistic categorizations of objects and services (e.g., the Universal Standard Products and Services Classification (UNSPSC) system).
7. **Privacy and Information Security**: It is vital for many potential uses of the GeoWeb infrastructure to ensure that privacy can be maintained in the face of a publicly accessible searchable infrastructure. A general framework is needed for restricting access to links, disallowing "tracking" of dynamic metadata items and allowing third parties to validate or endorse metadata entries.

GeoWeb Metadata

As the concepts that lead to the GeoWeb have developed, it has become clear that we have a somewhat different view of what may be considered georeferenced information and thus envision a very different kind of metadata than is traditional. Traditional metadata (e.g., Dublin Core, FGDC, or ISO/TC211) usually describe a particular, single data item or service (e.g., a single Web site, Web page, aerial image, or photograph). The searchable metadata we have been developing are different from this in a fundamental way: we wish to provide a semantic and geographical description of physical or conceptual objects that may have a number of different manifestations as data or services.

For example, a physical object such as a restaurant may have a metadata entry keyed to its location that contains a description and links for several distinct data elements (e.g., a home page, a menu, and a VRML model of the building) and services (e.g., a Voice over IP address, an online takeout order form, etc.). A user should be able to search for "a South Indian restaurant with an online menu and photographs within walking distance of my hotel." While it makes a great deal of sense to provide a semantic nexus for information at this level, this does not correspond to a traditional view of geographic metadata. Nevertheless, we have endeavored to build this level of capability on top of existing metadata work such as the ADL Collection Metadata (Hill et al, 1999).

It is worth noting, however, that the semantic break is not actually that great. It should be clear that traditional geographic metadata are actually encompassed by this broader conceptualization. A publisher could consider a single piece of data to be the appropriate conceptual object to be described by a metadata entry. It should be up to the metadata publisher to choose which level of object description is most appropriate for representing information within the index. Given that the choice to register a metadata entry is a very simple form of advertising, this flexibility can be considered to be one of the advantages of an opt-in registration strategy.

The GeoWeb DNS Hierarchy of Servers

Metadata in the GeoWeb are distributed across a number of servers for scalability. The servers have distinct Internet Domain Names that identify regions on the Earth bounded by latitude and longitude. Such a region is called a cell, and the domain name that identifies it is called a geographic domain name (see Figure 1). Following are example applications of geographic domain names using the minutes.degrees.tendegrees.geo name schema. (This schema is used here for simplicity, although the schema currently favored is actually of the form tenminutes.degrees.tendegrees.ninetydegrees.geo.)

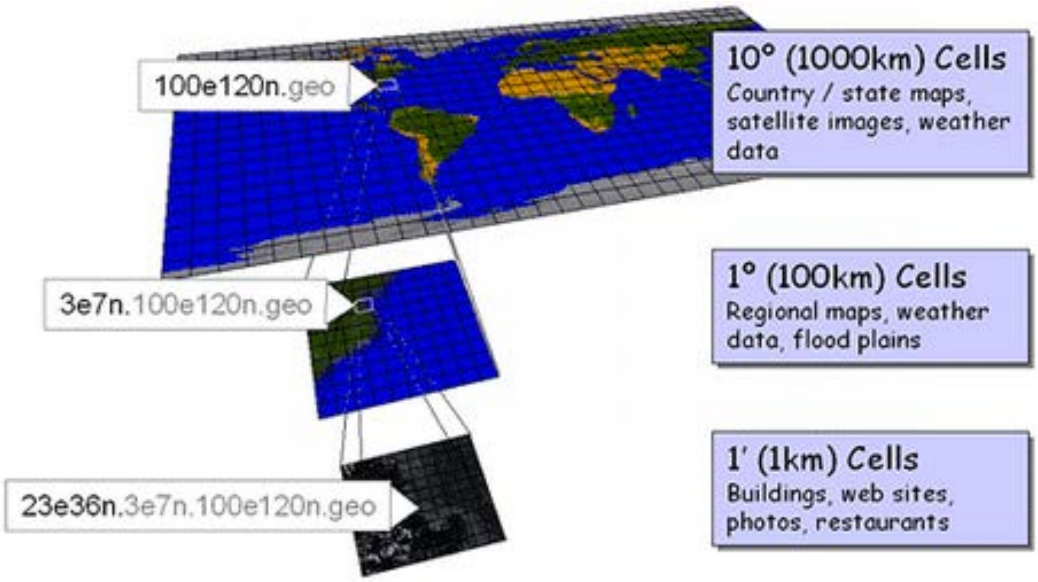


Figure 1. An illustration of the GeoWeb Heirarchy

- The geographic domain name *20e30n.geo* identifies the 10-degree x 10- degree cell whose southwest corner is located at 20 degrees east, 30 degrees north.
- The geographic domain name *2e4n.10e50n.geo* identifies the 1-degree x 1-degree cell whose southwest corner is located at 12 degrees east, 54 degrees north.
- The geographic domain name *11e21n.3e7n.30e10n.geo* identifies the 1-minute x 1-minute cell whose southwest corner is located at 33 degrees, 11 minutes east and 17 degrees, 21 minutes north .

Metadata are placed in the hierarchy according to the number of cells that overlap the geographic coverage of the data they correspond to (see Figure 2). Specifically, the level with the smallest cells is used so that no more than four cells overlap the geographic coverage. At this chosen level, the metadata are stored in all the overlapping cells.

This naming convention and metadata placement strategy allows clients to determine which host(s) to query for metadata referring to a given area, thereby distributing the load over many servers with no single point of failure or congestion. This distribution and query method also reduces the storage and bandwidth requirements of any given server, making it possible for thousands of high-speed visualization systems to be simultaneously searching for and retrieving metadata and data.

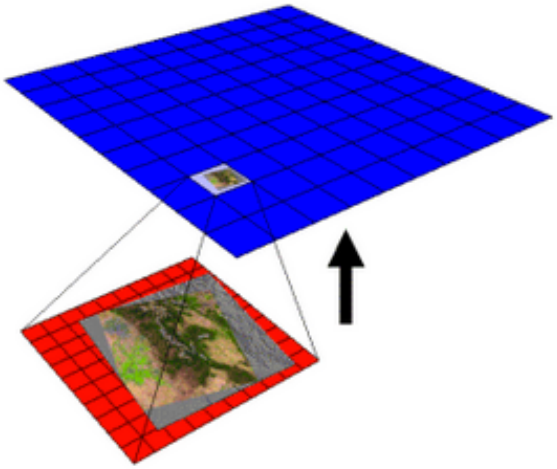


Figure 2. Demonstration of the procedure to locate the appropriate cell server(s) to store a new GeoWeb item.

The use of DNS also allows metadata to be transparently moved to new hosts and subdomains as needed. For example, initially all metadata may be physically hosted on a single computer, with all .geo domain names aliased to that machine. But over time, as areas fill with metadata (such as large urban areas), their corresponding subdomain and metadata can be transparently

transferred to new computers. These computers may be maintained by other organizations who may, when necessary, transfer data to new computers (and new organizations) deeper in the hierarchy.

Dynamic GeoWeb Content

Many objects that might gain some advantage by being registered in the GeoWeb do not have fixed positions at all, and it is their current position (and potentially history and planned future of movement) that should be reflected in the GeoWeb. These dynamic objects require a different interface to maintain their positions in the GeoWeb hierarchy and their associated geographic domain names. One must design a facility whereby an intermediate directory server can manage the metadata for such objects based on a globally unique object identifier instead of geographic location. This intermediate server then translates updated locations to the appropriate geographic domain name and updates the metadata in the GeoWeb hierarchy, taking care to remove metadata entries when they dynamically move out of particular cells. These intermediate servers may be either public or private (depending on the need to advertise or hide the object-ID to metadata mapping) and may offer other associated services. By structuring the solution this way, all of the distribution and localized discovery advantages of the GeoWeb architecture are retained while allowing dynamic objects to efficiently update their locations.

For example, an aircraft equipped with GPS receivers and secure wireless Internet access could periodically contact the dynamic directory server to update its current location, velocity, and other attributes. This information would then be used to update the geographic coordinates (and other attributes) in the corresponding metadata record in the GeoWeb hierarchy, and, when necessary, transfer the metadata record to a new cell. This dynamic metadata can then be used for many purposes, including air traffic analysis and providing global near-real-time maps of air traffic. Of course, the aircraft could in turn use the GeoWeb to download or access localized flight/airspace support information.

In the case of weather data, the dynamic directory server might be maintained by the U.S. National Weather Service itself, since its mandate includes the maintenance of real-time, public weather maps and storm information. As in the example above, the dynamic directory server would periodically update the corresponding metadata record and, when necessary, transfer it to a new cell.

Should the resource and network loads on these dynamic directory servers become too great for traditional single-point database methods, it would be relatively straightforward to take advantage of the strategy used by the location-based GeoWeb hierarchy and create a distributed DNS hierarchy to represent an object-identity index. For example, airplanes already have a registration ID that could be translated (and possibly encrypted for privacy) to a DNS name/id pair such as*us.air.geo/AC7321*. This server/id pair could then be used with exactly the same protocol as is used to update GeoWeb registry cells to update a dynamic directory server which then automatically updates the associated entry in the location-based GeoWeb hierarchy. In database terms, we can provide a second primary index into the global metadata database using exactly the same update protocols. These secondary indices could be either private or public, depending on data privacy concerns in the particular domains implemented.

Validation of Metadata and Data

To ensure the accuracy and validity of certain classes of data for certain purposes (such as elevation data used for city planning purposes), the metadata record has a field for one or more optional validation certificates issued by validation organizations. The digitally signed certificate will certify that one or more elements of the metadata/data meet certain qualifications, such as accuracy or completeness, or compliance with local, national, or international standards or conventions. While essential for validation purposes, this facility also has a number of other uses, such as for reviews. For example, an organization such as the American Automobile Association could annotate hotel and restaurant listings with references to their own independent descriptions of the facilities and quality of service that these businesses provide.

Clearly this facility has further implications for security and privacy than those already outlined above.

GeoWeb Advantages

The GeoWeb infrastructure offers a number of advantages over today's state-of-the-art technology. It is specifically designed as an extremely scalable, open, and global geographical index to the Web. Below are a few of the key benefits of this technology.

- **Open opt-in scheme**. No one company or institution can hope to index all geographically related information for the whole planet. Therefore, the GeoWeb is an opt-in scheme where users can register their own data, be it information about their bed and breakfast, or pictures of their trip to the Big Island.
- **Integrate disparate data sources**. Instead of being restricted to one company's view of the world, the GeoWeb allows users to discover and integrate disparate sources of information for improved decision-making.
- **Massively scalable architecture**. The entire GeoWeb index could be stored on one server, but as demands and database size grow individual cell servers can be split off on to additional physical machines. To illustrate the inherent scalability, there are over 60,000,000 1-minute servers possible over the land regions of the earth.
- **No one-server bottleneck**. Most contemporary search and indexing interfaces require going through a single point-of-failure server. In the GeoWeb, the client works out which cell server to contact based on the geographic extent of the query.
- **Transparent load balancing**. Using the DNS hierarchy enables transparent load balancing by moving data to new servers and organizations as needed. The IP addresses change and the bandwidth improves, but the domain names remain the same.
- **Index any content**. The GeoWeb can index any content, not just HTML. Movies, sound files, text, 3D models, terrain, etc. can all be indexed. Content such as HTML with geographic meta tags allows an easy way to register that content in the GeoWeb.

- **Built-in security**. Security is imperative to ensure that private data remains private, while still providing easy access to public information. In addition, there is a scheme to allow appropriate organizations to validate and/or endorse data.
- **Support high bandwidth demands**. Massive distribution of the geographic database enables many clients around the world to retrieve data simultaneously and at high speed.
- **Reduced server costs**. As the GeoWeb infrastructure is distributed over many cell servers, there is no need for a single massive server with huge storage requirements. The size and speed needed per server is therefore reduced, for example, divided by 10,000 or more for 10-minute servers.
- **Local control**. Companies, institutions, or local governments can manage the cell servers for their region of the world, and also host the cell servers in their geographic region.

GeoWeb Clients

Naturally, users will not normally browse the GeoWeb by typing in domain name addresses such as*11e21n.3e7n.30e10n.geo*. The benefit of the GeoWeb is that a client application or portal can trivially work out the server to contact for any given region of the earth. A range of possible GeoWeb clients can be envisioned. The following sections introduce a few of the interfaces that are proposed or are already under development.

Text-based: The most basic kind of interface would be similar to the types of search engines that we use today. However, in addition to entering a search criterion, you would also enter a location for your search. This could be a latitude/longitude coordinate, or more intuitively, a street address or a feature name that can be translated to a geodetic coordinate using a gazetteer service. The results could be presented as a simple list of links (ordered by distance), as is the case for most search engines today. This type of interface has the advantage that it scales down well to current handheld devices such as a PalmPilot &#153 or iPad &#153, as shown in Figure 3(a).

Map-based: There are already various mapping sites on the Web that will generate a 2D map for a region of interest, e.g., MapBlast!, Yahoo! Maps, MapQuest, Maps.com, etc. It is easy to imagine a GeoWeb client that lets users search over a particular geographic region and return the results as a map image with various icons overlaid (see Figure 3(b)). These icons could be hyperlinked so that clicking over them would take you to more information about that item, for example, clicking over the icon for a restaurant might take you to their menu or on-line reservation page.

TerraVision: TerraVision &#153 is a distributed, interactive terrain visualization system developed by SRI International. It allows users to navigate, in real time, through a 3D graphical representation of a real landscape created from elevation data and aerial images of that landscape. TerraVision is described in detail later in this chapter. It is designed as a GeoWeb client so that, as the user flies around the world, the system is constantly querying the GeoWeb index to discover all relevant information about the place being viewed. This information is then streamed automatically to the user's display for browsing.

Geoster: The popularity of file sharing protocols and applications has been amply demonstrated with the enormous usage of services like Napster and Gnutella. SRI is developing a file sharing application along the lines of Napster, but using the GeoWeb as the underlying indexing and search infrastructure. Geoster (pronounced *jester*) is an application built for sharing photos over the Web using the GeoWeb (see Figure 4). It lets users index photos from around the world and to share these with the rest of the world. This would allow users to do searches such as, show me all photos that have been taken of the Great Wall of China. Geoster is freely available fromhttp://www.geoster.com/ and is written in Java and hence available across multiple platforms. This type of client could produce a large interest in the GeoWeb and generate a substantial amount of GeoWeb registrations.

Figure 4. A sample GeoWeb client: the Geoster photo-sharing client.

The GeoVRML Standard

Dealing with 2D maps and simple feature overlays has long been a common task for traditional Geographic Information Systems (GISs). However, enabling interactive, accurate, and dynamic 3D visualizations of geospatial data that can be disseminated over the Web is only now becoming a reality. The GeoVRML working group of the Web3D Consortium recently announced an extension to the ISO standard Virtual Reality Modeling Language (VRML, 1997) to enable just these capabilities. This extension is called GeoVRML 1.0 (Reddy et al., 2001; Reddy et al., 2000a; Reddy et al., 1999b). GeoVRML is an official recommended practice of the Web3D Consortium and is included in an upcoming amendment to the VRML97 ISO standard.

GeoVRML 1.0 content can be browsed interactively in any standard VRML97 browser. A number of these browsers exist as freely available plug-ins for common Web browsers such as Netscape Communicator and Microsoft Internet Explorer. Tools have been written to simplify

the authoring and conversion of geospatial data to GeoVRML. In addition, GeoVRML is an open standard: its specification is published openly and a source-level sample implementation is provided. In this section, we will concentrate on a few of the commercially available tools that support the GeoVRML format, and also describe some of the capabilities that this solution provides geoscientists for the purpose of integrating their geographic data directly into a 3D computer graphics scene graph.

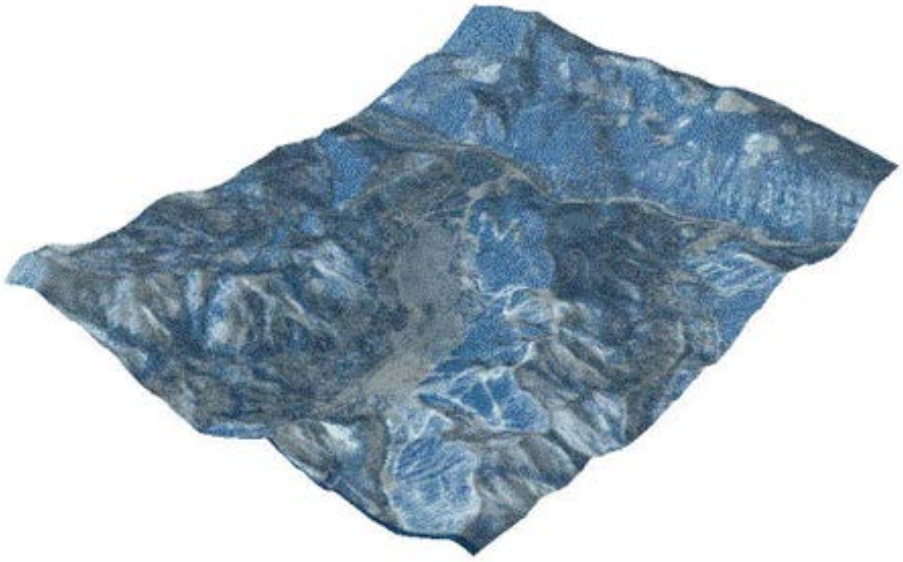


Figure 5. A GeoVRML model of Squaw Valley, CA, output by ESRI's ArcInfo 8.1 product (with 3D Analyst Extension). Model provided courtesy of ESRI.

Tools

A number of companies have developed tools that support the GeoVRML file format. These include modeling tools that can export to GeoVRML, conversion tools to take standard mapping products and produce GeoVRML representations of these, and visualization technologies that allow the user to browse GeoVRML content. The follow sections present a selection of these tools. More examples can be found from the GeoVRML home page (http://www.geovrml.org/).

ArcInfo / ArcView: With version 8.1 of the 3D Analyst extension for the ArcView and ArcInfo products, released in April of 2001, ESRI added GeoVRML support to their range of capabilities. These products now allow export of GIS data to the GeoVRML file format for viewing over the Web. Figure 5 was created with ArcInfo 8.1 by combining an orthographic image (retouched to have a blue color) with a DEM of the Squaw Valley ski resort, near Lake Tahoe. The ESRI home page ishttp://www.esri.com/.

ShapeViz: Bashir Research produces a tool that allows you to view Shape files. This file format was created by ESRI as a way to interchange geometry data and is supported by their range of GIS products. ShapeViz can take Shape files and convert these to VRML for interactive visualization over the Web. With version 1.2 of ShapeViz, GeoVRML export support was added, along with the ability to specify the coordinate system of the Shape file (as this information is not ordinarily included in the Shape file). Figure 6 shows an example Shape file converted to GeoVRML using the ShapeViz tool. The Bashir Research home page is http://www.my3d.com/.

Cortona: ParallelGraphics produce a popular VRML browser caller Cortona, which is provided as a plug-in for Netscape and Internet Explorer. With version 3 of the Cortona browser, ParallelGraphics added a native implementation of the GeoVRML extensions. The implementation for these nodes can be automatically downloaded and installed when GeoVRML nodes are first used. The original GeoVRML implementation is written in Java so that it can run within any VRML browser that supports Java as a scripting language. With a native implementation, the Cortona browser offers more integrated and efficient support for this geospatial format. Figure 6 shows the converted Shape file being viewed with the Cortona browser. The Cortona browser can be found at http://www.parallelgraphics.com/.

Figure 6. A GeoVRML translation of a Shape file produced by the ShapeViz utility from Bashir Research. This shows a number of layers for Mexico, including state boundaries, rivers, roads, lakes, and cities. This is being displayed in ParallelGraphics' Cortona 3.1 browser. Model provided courtesy of Bashir Research.

DEM2GeoEG: DEM2GeoEG is a program to convert USGS

Digital Elevation Model (DEM) data into a VRML .wrl file that uses the GeoVRML 1.0 GeoElevationGrid (see below). The GeoElevationGrid is a version of the standard ElevationGrid but it lets you georeference the data to geographic coordinate systems such as UTM (Universal Transverse Mercator). One benefit of this is that you can fuse multiple GeoElevationGrids into a single scene and they will be correctly located with respect to each other. Figure 7 illustrates a USGS DEM converted to GeoVRML using this utility. The program and source code can be found from the GeoVRML home page at http://www.geovrml.org/.

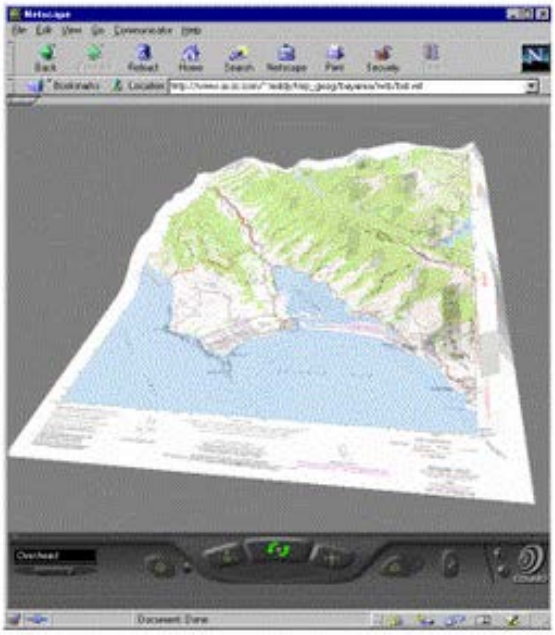


Figure 7. A USGS DEM that has been converted to a GeoElevationGrid using the dem2geoeg utility, and then draped with a DRG image for the same area.

tsmApi: The Tile Set Manager Application Program Interface (tsmApi) from SRI International is an Open Source library of high-level C functions for reading, writing, and processing the terrain data used by the TerraVision terrain visualization application. The tsmApi library can ingest a number of standard mapping products and produce GeoVRML representations of these terrain models. Specific formats that are supported include USGS DEM and DOQ, DTED level 0, GeoTIFF, TIFF, GIF, JPEG, PPM, and PGM. For further details, refer to http://www.tsmapi.com/.

Capabilities

The following list provides a high-level description of the capabilities that are specifically addressed by GeoVRML 1.0.

1. **Geographic Coordinate Systems**. Most 3D graphics systems, including VRML, use a simple Cartesian (x,y,z) coordinate system to model all data. GeoVRML provides the ability to also specify locations using geodetic or projected coordinate systems such as

those commonly used in the geosciences. Specifically, GeoVRML 1.0 has support for geodetic (latitude/longitude), Universal Transverse Mercator (UTM), and geocentric coordinate systems. In addition, 21 ellipsoids and 1 geoid are supported. This support is built upon the SEDRIS Spatial Reference Model (see http://www.sedris.org/).

2. **Data Fusion**. GeoVRML can handle data from disparate servers across the Web, generated from different sources, at different resolutions, and specified in different coordinate systems. These are all fused into a single global context for visualization. For example, you can overlay a Global Positioning System (GPS) track of latitude/longitude coordinates over a UTM-rectified terrain model.
3. **High Precision**. VRML97 provides only single-precision floating-point values. This is insufficient to represent data on a planetary scale down to around meter resolution or beyond. GeoVRML provides solutions to extending this precision to sub-millimeter by employing the specification of local coordinate systems.
4. **Dataset Scalability**. Terrain models often involve large elevation grids or image files. GeoVRML provides various scalability features to manage the streaming of large, multi-resolution models, thus facilitating real-time access and display of arbitrarily large terrain models.
5. **Metadata Linking**. GeoVRML provides the ability to specify a generic subset of metadata describing geographic objects, including the ability to link to a full metadata description.
6. **Animation Support**. The ability to perform key frame animations within the supported geographic coordinate systems is provided so that animations can be defined with respect to key points on the surface of the planet.

Implementation

Essentially, GeoVRML 1.0 consists of ten new extensions, or nodes, that sit on top of VRML97, i.e. GeoVRML includes all of VRML97 as a subset. These nodes are defined using VRML's EXTERNPROTO extensibility features. For the purposes of the initial sample implementation, these nodes have been implemented using Java class files that are embedded within the new

nodes; although we have already noted that at least one commercial VRML browser has now implemented these nodes natively. The following sections detail a small selection of the nodes that are provided by GeoVRML 1.0 (Reddy et al., 2000b).

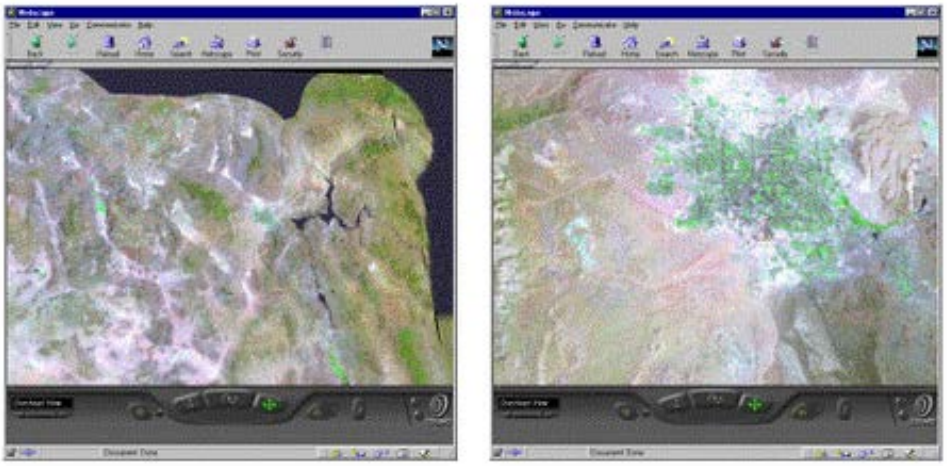
GeoCoordinate: The GeoCoordinate node enables the specification of coordinates by using geographic coordinate systems. This node can be used within standard VRML geometry nodes such as IndexedFaceSet, IndexedLineSet, or PointSet, enabling the modeler to specify coordinates in a system such as UTM. For example, a GPS will normally output location as a latitude/longitude coordinate. With the GeoCoordinate node, we can insert these coordinates directly into a VRML file and have them integrated with any other geospatial data.

GeoElevationGrid: The GeoElevationGrid node provides the capability to define a grid of height values offset from the ellipsoid or geoid used to model the planet. It supports the specification of height fields in latitude/longitude or UTM coordinate systems. VRML97 already provides an ElevationGrid node; however, in this node all values are offset from a single flat plane. This is acceptable if the area being modeled is small in extent, for example, less than 1 km2, but for larger areas the curvature of the earth becomes significant. Figure 7 illustrates an example showing a GeoElevationGrid.

GeoLocation: The GeoLocation node lets the user georeference an arbitrary VRML model, that is, locate it at a specific point on the earth. It also orients the model correctly, depending upon its position on the earth, so that +Y is aligned with gravitational up, +Z points true north, and +X points east. This ensures that a model built using the standard VRML right-handed coordinate system will be placed on the earth so that its base is aligned with the surface of the planet. Figure 8 illustrates the capabilities of the GeoLocation node by georeferencing models of individual buildings of the SRI International main campus to an underlying terrain model of Menlo Park, CA.

Figure 8. A texture-mapped site model of the SRI International campus that has been georeferenced onto an underlying terrain model. Note the accurate alignment of the latitude/longitude-specified buildings with the underlying 1 m resolution USGS DOQ satellite imagery in UTM coordinates.

GeoLOD: The GeoLOD node provides the capability to browse multi-resolution, tiled terrain data that are streamed over the Web. It automatically manages the progressive loading of higher-resolution data as the user approaches the terrain, and also unloads terrain data that the user has flown past. These are essential memory management and scalability operations for browsing massive terrain datasets. For example, Figure 9 illustrates a large multi-resolution dataset that is built using the GeoLOD node. We illustrate the capability to fly down through several levels of detail while higher-resolution data are streamed over the Internet to the user's display.



(a) (b)

Figure 9. An example showing the scalability features provided by the GeoLOD node. The two images show a user's descent into a terrain model of the Mojave desert in southern California. The source imagery and elevation data for this terrain model total 1.3 GB.

GeoPositionInterpolator: One of the many strengths of VRML is its ability to model dynamic systems. GeoVRML therefore incorporates the capability to animate models using geographic coordinates. This is implemented through the GeoPositionInterpolator node, which functions in much the same way as the standard VRML97 PositionInterpolator node, except that the key frame values can be specified using geographic coordinates. For example, if a GeoPositionInterpolator is created and given two geodetic coordinates, (122.413 W, 37.7483 N, 10000) and (116.388 E, 39.906 N, 10000), and the output is routed to a VRML model of a Boeing 777, then this aircraft would fly over the surface of the planet, from San Francisco to Beijing, at a constant altitude of 10,000 m. Using this capability, a vehicle could be animated based upon the list of latitude/longitude coordinates from a prerecorded GPS track, or from a live feed of Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs).

GeoVRML Standards Positioning

GeoVRML 1.0 is currently an accepted recommended practice of the Web3D Consortium. In addition to this, the GeoVRML nodes have been submitted to ISO as part of an amendment to the VRML97 specification. The intention is that these capabilities will eventually become an

optionally implementable part of the current VRML ISO specification, and also an additional profile in any future revisions of the specification.

The GeoVRML work is of major relevance to several exciting initiatives that are currently evolving. For example, this work provides the technical foundation to build the vision of a highly accurate and large-scale model of the earth into which we can embed massive quantities of georeferenced data: a vision that is sometimes referred to as the Digital Earth (Leclerc et al., 1999; Reddy et al., 1999b). Seehttp://www.ai.sri.com/digital-earth/ and http://www.digitalearth.gov/.

On a related topic, the OpenGIS Consortium (OGC) is evolving its Web Mapping Testbed (WMT) technology as a means to revolutionize the use of geospatial data on the Web (de La Beaujardiere, 2002). Its efforts to date have focused largely on 2D presentations and also on the cataloging of geographic data. As such, the GeoVRML work is well positioned to integrate with the OGC efforts and add Web-based 3D visualization capabilities to this initiative; for example, SRI has produced an OGC-compliant Web Map Server (WMS) that can produce results in GeoVRML format as well as the more common GIF and JPEG image formats.

Additionally, the GeoVRML working group is contributing to the evolving X3D development. X3D is the next-generation VRML specification and supports extensibility through a number of profiles that can be plugged into a core browser (Daly and Williams, 2002; Bietler et al., 2002). The GeoVRML nodes have already been added as a Geospatial profile for X3D and a document type definition (DTD) for this profile has been developed. For further information, see the X3D Web page athttp://www.Web3d.org/x3d.html.

Finally, the GeoVRML group is currently compiling issues for a future version of GeoVRML, including support for more coordinate systems, ellipsoid definitions for all other planets in the solar system, new GeoVRML nodes, and additions to various current nodes.

The TerraVision System

TerraVision ™ is a system for interactively browsing 3D representations of large geographic areas (Leclerc and Lau, 1994; Reddy et al., 1999). It was developed in response to shortcomings in other systems that can only browse terrain data from local disk resources, and hence are limited by users' ability to store and maintain entire datasets locally. TerraVision can therefore retrieve and merge massive volumes of remotely located data, including aerial and satellite imagery, topography, weather data, buildings, and other cultural features (see Figure 10). These data can be terabytes in size, distributed over multiple servers across the Web, and can be automatically discovered using the GeoWeb technology described earlier in this chapter.

Figure 10. Screenshot of the TerraVision system.

User Scenario

The user starts TerraVision and sees a 3D view of the Earth. As the user travels around the globe, browsing for a particular area, TerraVision automatically fetches data distributed across many sites throughout the Internet by querying the GeoWeb. The user flies down to ground level and is able to find all nearby restaurants, places of interest, and even the local garage sales on that day.

Performance, scalability, and portability make TerraVision (and its associated data) a useful tool for numerous applications, such as military personnel performing mission planning and battle damage assessment, emergency teams fighting a forest fire or organizing disaster relief efforts, environmental workers evaluating a floor, or other time-critical conditions.

Technology Overview

To interact with massive, remotely located repositories in real-time, TerraVision employs a tiled, multi-resolution data representation (described in detail below). This involves segmenting the original data into rectangular tiles over a range of resolutions, where each tile contains the same number of pixels or elevation data. By employing customized caching, culling, and data fetching optimizations, the number of polygons and texture maps required for rendering remains approximately constant, independent of dataset size and the viewpoint. Tiles are requested for an area using a coarse-to-fine progression so that TerraVision always has low-resolution data for the area of interest. TerraVision supports the open standard GeoVRML format for representing building, weather, and other 3D entities. TerraVision handles multiple types of imagery; allowing the user to select or blend between different datasets, e.g., full aerial and weather Earth models where parts of the surface will have imagery down to 1 m resolution. The user can navigate the terrain data with standard software on a personal computer connected to the Internet. A user can employ a graphics workstation connected to a fast network with high-speed disk servers to quickly navigate around a large area, but can also access this same data from a laptop machine over a wireless link when working in the field.

TerraVision Features

- **Distributed Data**. TerraVision can browse data that is distributed over a wide- area network, e.g., the Web, as well as locally installed data. TerraVision was specifically designed to cope with the inherent unpredictability of accessing data over a network.
- **Massive, Scalable Datasets**. TerraVision can view massive datasets, in the order of terabytes. It achieves this by employing powerful optimization algorithms including: view frustum culling, terrain and imagery level-of-detail, horizon culling, caching, and prediction.
- **Entire Earth Visualization**. TerraVision can handle datasets in a variety of geographic coordinate systems, e.g., lat/long, UTM, LVCS, and can transform these on the fly to a round-earth, or geocentric, representation.
- **Multiple Datasets**. TerraVision can view multiple datasets at once. For example, you can have a 1 km resolution globe model, with a 25 m model of the San Francisco Bay Area, and 1 m inset for Palo Alto.
- **Multiple Viewers**. TerraVision allows you open up multiple viewer windows. This lets you look at the same dataset from different perspectives at the same time, or different combinations of sets. You can even control the view of one viewer through another.
- **GeoVRML Model Overlays**. 3D models can be overlaid on the terrain to provide support for cultural features, such as buildings and roads, and atmospheric simulations, such as wind vectors and clear air turbulence models. We use GeoVRML to represent all models with a precise geographic location.
- **OGC WMS Support**. The OpenGIS Consortium (OGC) developed the Web Mapping Server (WMS) interface as a standard way to serve and integrate maps over the Web. TerraVision can access data from any OGC-compliant WMS, and we include an OGC-compliant WMS Apache module to serve TerraVision and GeoVRML models.
- **Flight Paths**. You can set up predefined flight paths by marking a number of viewpoints and then telling TerraVision to fly a path connecting those viewpoints. You can vary the velocity, loop the path, close the loop, and select linear or spline interpolation.

- **Viewpoint Bookmarks**. If you like a particular viewpoint, for example if you have found your house, then you can bookmark that viewpoint and TerraVision will remember it so that you can fly back to it later, or next time you use TerraVision.
- **Heads Up Display**. A simple HUD is available to provide information such as viewer location (in lat/long), orientation, number of tiles display, data burst rate, and frame rate.
- **Documentation**. A Web-based user guide is available to help new users familiarize themselves with the TerraVision system. TerraVision offers Help menus that bring up appropriate sections of this user guide in their Web browser.
- **Cross platform**. TerraVision is available for SGI IRIX (6.3 and up), Linux, and Windows 98/ME/NT/2000. It is available as a full version with Tcl/Tk interface, a Netscape plug-in, and an ActiveX component for embedded in Microsoft applications such as Internet Explorer and PowerPoint.

Managing Multiresolution Grids

Level of detail (LOD) is the technique of changing a model's complexity based upon some selection criteria, such as distance from the viewpoint or projected screen size (Luebke et al., 2002). The basic premise for these two selection criteria is that any distant detail that projects to less than a single pixel on the screen will not generally be visible. In order to implement this, we need a mechanism to simplify the geometry and imagery for a dataset. Several polygon simplification algorithms exist that work well for terrain. However, many of these are view independent techniques that force the same degree of simplification across the entire terrain. These are not appropriate for our application because switching to the highest resolution would still involve loading every point of the original dataset. Instead, we require a view independent technique where the degree of simplification can be varied with respect to the current viewpoint. This is often done using a hierarchical data structure, such as a quad-tree. A further requirement is that the LOD algorithm must not require access to the entire high-resolution version of the dataset. This is necessary because we do not want to be limited to viewing only

datasets that can fit onto the user's local storage system. These requirements suggest that a tiled, pyramid representation is best suited to our needs.

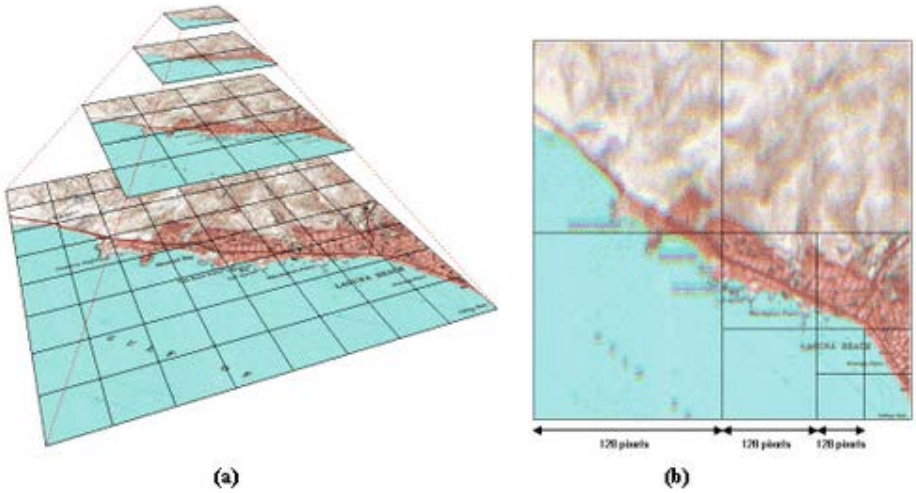


Figure 11. An example image pyramid. (a) Illustrates four different resolutions of an original image, where each level of the pyramid is segmented into 128 x 128 pixel tiles. (b) Shows how this structure can be used to alter the resolution of an image in different regions.

A pyramid is simply a multi-resolution hierarchy for a dataset (DeFloriani, 1989). For example, if an original image is 1024 x 1024 pixels at 1m resolution, then an example pyramid might contain this original image along with down-sampled versions at resolutions of 512 x 512 pixels (2m), 256 x 256 pixels (4m), 128x 128 pixels (8m), and so on. Each image of the pyramid is then segmented into rectangular tiles, where all tiles have the same pixel dimensions, e.g., 128 x 128 pixels as in Figure 11(a). A tile at one level of the pyramid will therefore map onto four tiles on the immediately higher resolution level. That is, the tiles at the higher resolution level cover half the geographical area of the former.

Using this representation, it is possible to recursively resolve certain regions of a dataset in more detail than other regions. For example, in Figure 11(b) we have displayed the lower-right corner in high resolution with the surrounding regions displayed in progressively lower resolution. Assuming a tile size of 128 x 128 pixels, this example requires downloading and rendering only 491 KB (10 tiles) instead of the entire 3.1 MB high-resolution image. If the user is

located at the bottom-right corner, then we gain the effect that distant imagery is rendered at lower resolution than near imagery, i.e. we have implemented distance-based LOD.

For simplicity, we have only considered pyramids of images so far. However, the same techniques can be applied to elevation grids, as well as other types of terrain data. By employing a tiled pyramid representation for the geometry as well as the imagery, TerraVision is able to optimize the amount of data transferred over the network, the number of polygons in the scene, and the amount of memory required for texture maps. The effect of this technique is that we only need to fetch and display data for the region that the user can see, and only at a sufficient resolution for the user's viewpoint. This solution scales well to arbitrarily large datasets because it effectively attempts to keep the polygon count constant for any viewpoint.

Summary

Three tools have been developed to solve the problems of discovering, modeling, and visualizing global grids and associated data.

As a fundamental new service on top of the Web, the GeoWeb provides a unique solution for harnessing the power of the Internet. It is based on the way human beings perceive and comprehend their world: geospatially, in three dimensions, and over time. The GeoWeb enables Internet users to navigate, access, and visualize georeferenced data as they would in a physical world, but without the barriers imposed by space and time in the physical world (see http://www.dotgeo.net/).

GeoVRML provides the geosciences field with a rich suite of enabling capabilities that cannot be found elsewhere as a Web browser plug-in. That is, the ability to model dynamic 3D geographic data that can be distributed over the Web and interactively visualized using a standard browser configuration. With the presence of this standard, and the recent emergence of many low-cost 3D accelerator cards for common desktop computers, the scene is now set to allow

researchers, educators, and businesses to publish their rich, 3D geospatial products to a wide audience over the Web (seehttp://www.geovrml.org/).

TerraVision provides a freely available and cross-platform capability to browse massive terrain datasets in 3D. It can browse huge datasets, in the order of terabytes, and these data can be distributed over multiple servers across the Web. It is available as a full application, a Netscape plug-in, and an ActiveX component. The system supports the overlay of 3D GeoVRML models and offers an OGC-compliant Web Mapping Server interface. The tools to generate TerraVision datasets are released as Open Source for anyone to download and augment (see http://www.tvgeo.com/).

References

M. Bietler, M. Bourges-Sevenier, D. Brutzman, P. Diefenbach, T. Parisi, M.Reddy and D. Silverglate (2002). "X3D Architecture and Profiles". Tutorial at the 7th International Conference on 3D Web Technology, February 24-28, 2002.

L. Daly and J. Williams (2002). "Introduction to X3D". Tutorial at the 7th International Conference on 3D Web Technology, February 24-28, 2002.

L. DeFloriani (1989). "A Pyramidal Data Structure for Triangle-based Surface Description". IEEE Computer Graphics and Applications, 9(2), pp. 67-78.

L. L. Hill, G. Janee, R. Dolin, J. Frew and M. Larsgaard (1999). "Collection Metadata Solutions for Digital Library Applications", Journal of the American Society for Information Science, 50(13): 1169-1181.

Y. Leclerc, M. Reddy, L. Iverson, and M. Eriksen (2001a). "The GeoWeb (aka dot-geo) --- Indexing Data on the Internet by Location". Digital Earth 2001, New Brunswick, Canada, 24- 28 June 2001.

Y. Leclerc, M. Reddy, L. Iverson, and A. Heller (2001b). "The GeoWeb --- A New Paradigm for Finding Data on the Web". In Proceedings of the International Cartographic Conference (ICC2001), Beijing, 6-10 August 2001.

Y. Leclerc, M. Reddy, L. Iverson, M. Eriksen, and A. Heller (2000). "The .geo-web: A Scalable Index for the Digital Earth". In Proceedings of GIScience 2000, Savannah, Georgia. October 28-31, 2000

Y. Leclerc, M. Reddy, L. Iverson, N. Bletter (1999). "Digital Earth: Building the New World", In proceedings of the 5th International Conference on Virtual Systems and Multimedia (VSMM'99), pp. 250-262. 1-3 September. Dundee, Scotland.

Y. G. Leclerc and S Q Lau (1994). TerraVision: A Terrain Visualization System. Technical Report Technical Report 540. SRI International. Menlo Park, CA. April 1994.

D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson and R. Huebner (2002). "Level of Detail for 3D Graphics", Morgan Kaufmann Publishers, San Francisco, CA.

J. de La Beaujardiere (Editor). "OpenGIS Web Map Server Interface Implementation Specification," 16 Jan 2002. http://www.opengis.org/techno/implementation.htm

M. Reddy, L. Iverson, Y. Leclerc, and A. Heller (2001). "GeoVRML: Open Web-based 3D Cartography". In Proceedings of the International Cartographic Conference (ICC2001), Beijing, 6-10 August 2001.

M. Reddy, L. Iverson, and Y. G. Leclerc (2000a). "Under the Hood of GeoVRML 1.0". In Proceedings of Web3D-VRML 2000: The Fifth Symposium on the Virtual Reality Modeling Language. Monterey, California. February 21-24, pp. 23-28.

M. Reddy, L. Iverson, and Y. Leclerc (2000b). "GeoVRML 1.0: Adding Geographic Support to VRML". GeoInformatics, Volume 3, September 2000.

M. Reddy, Y. G. Leclerc, L. Iverson, N. Bletter, and K. Vidimce (1999a). "Modeling the Digital Earth in VRML". In Proceedings of SPIE - The International Society for Optical Engineering. Volume 3905, pp. 113-121.

M. Reddy, L. Iverson, Y. G. Leclerc (1999b). "Enabling Geographic Support in Virtual Reality Modeling with GeoVRML", Cartography and Geographic Information Science. 26(3):180.

M. Reddy, Y. G. Leclerc, L. Iverson, and N. Bletter (1999). "TerraVision II: Visualizing Massive Terrain Databases in VRML". IEEE Computer Graphics and Applications (Special Issue on VRML), 19(2): 30-38.

The Virtual Reality Modeling Language. International Standard ISO/IEC 14772-1:1997. 1997.

The work performed by SRI International was funded in part by the Defense Advanced Research Projects Agency (DARPA) under contracts MDA972-97C-0037, subcontract 12165SRI of contract no. F19628-95-C- 0215), and MDA972-99-C-0011. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International.

1. The work performed by SRI International was funded in part by the Defense Advanced Research Projects Agency (DARPA) under contracts MDA972-97C-0037, subcontract 12165SRI of contract no. F19628-95-C-0215), and MDA972-99-C-0011. The views and conclusions contained in this document are those of the authors and should

not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International.

Chapter Four:

The Global Spatial Data Model (GSDM)

By Earl F. Burkholder, PS, PE

Surveying Engineering Department - 3SUR

New Mexico State University - Las Cruces, NM

Burkholder, E. F. (2000). The Global Spatial Data Model. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis

Abstract

Spatial data are 3-dimensional (3-D) and modern measurement systems collect data in a 3-D environment. Computer databases store 3-D digital spatial data. Human perception of spatial relationships is primarily visual and, due to gravity, our natural reference for spatial perception is horizontal (2-D) and vertical (1-D). Various models are used to establish a conceptual connection between the measurements, digital spatial data, and its representation - data visualization. Digital spatial data are also used to make analog products such as maps, charts, and other hardcopy diagrams. The point is, once spatial data are put into digital form, they can be manipulated at the whim of the user. To preserve the integrity and value of spatial data, there should be a common storage format and the data manipulation processes should be well-defined, unique, bi-directional, and three-dimensional. Of course, the challenge is to identify a common format for 3-D data that is at the same time simple, rigorous, and global. The data storage format must also accommodate reliable statistical measures of spatial data accuracy.

The global spatial data model (GSDM) described in this presentation is a collection of concepts and procedures which can be used to collect, organize, store, process, and manipulate 3-D spatial data. The GSDM uses one set of solid geometry equations which are equally applicable around the world. This simple standard preserves global interoperability and each discipline,

agency, corporation, or individual spatial data user is free to implement any derivative use or application.

The GSDM includes both a functional model and a stochastic model. The functional model encompasses the geometry of spatial relationships and the stochastic model defines the process for establishing, tracking, and reporting the accuracy of spatial data using existing standard mathematical procedures.

Introduction

An appropriate introduction to the global spatial data model (GSDM) might be to list several challenges facing geo-spatial data users. According to Vice President Al Gore (1998) in a speech titled, The Digital Earth: Understanding our planet in the 21st century, "the hard part of taking advantage of this flood of geo-spatial information will be making sense of it - turning raw data into understandable information." The following challenges all come under the umbrella of Gore's statement, but are listed separately for purposes of this discussion. Challenges for spatial data users include:

1. Handling vast amounts of 3-D digital spatial data efficiently.
2. Describing spatial data accuracy without ambiguity.
3. Finding the best (appropriate) combination of tools, talents, and resources to accomplish the task at hand.

The third challenge is very open ended and relies heavily on the judgment of the user but the first two challenges are addressed specifically by the GSDM and supported by an underlying BURKORD database (see www.zianet.com/globalcogo). The functional model portion of the GSDM includes geometrical equations that permit each user to work with local rectangular (flat earth) coordinate differences while preserving true geometrical integrity on a global scale. The stochastic model portion of the GSDM includes rigorous error propagation procedures that

accommodate input of measurement uncertainties and provide output of standard deviations for each 3-D coordinate position and/or other derived quantities. It has been said, "No job is difficult if you have the right tools." The goal of this paper is to examine features of the GSDM with the idea of finding better tools for handling digital spatial data. If the GSDM is an appropriate tool, and if it can be used beneficially by various disciplines, then the larger issue is becoming more familiar with the fundamental concepts, using and building on those concepts, and sharing that knowledge with others. As Al Gore concluded in his speech, "Working together, we can help solve many of the most pressing problems facing our society..."

Definition of the Global Spatial Data Model (GSDM)

The GSDM, formally defined by Burkholder (1997), is a collection of existing mathematical concepts and procedures that can be used to manage spatial data both locally and globally. It consists of a functional model that describes the geometrical relationships and a stochastic model that describes the probabilistic characteristics of spatial data. The functional part of the model includes equations of geometrical geodesy and rules of vector algebra (solid geometry) as related to various coordinate systems (see Figure 1, Diagram Showing Relationship of Coordinate Systems). The primary coordinate system used by the GSDM is the earth-centered earth-fixed (ECEF) geocentric X/Y/Z coordinate system as defined by the Department of Defense (DMA 1991). The GSDM is intended to be consistent with the 3-D Geodetic Model described by Leick (1995) with the following exception; the GSDM, being strictly spatial, does not include gravity measurements, but presumes the effect of gravity is accommodated before data are entered into the spatial model.

The stochastic model component of the GSDM uses fundamental error propagation concepts as described in Chapter 4 of Mikhail (1976) and Chapter 5 of Wolf/Ghilani (1997). The GSDM stores stochastic information in the variance/covariance matrix associated with each point defined by ECEF coordinates and in the correlation between point-pairs. The local perspective

(e/n/u) covariance values and standard deviations need not be stored but are computed upon demand from the geocentric values. The accuracy defined by each point covariance matrix, whether geocentric or local, is called **datum accuracy**. A BURKORD 3-D database accommodates storage of both the geocentric covariance values for each point and point-pair correlation. As described by Burkholder (1999), that capability supports and allows one to exploit the rigorous mathematical definitions of **local accuracy, network accuracy**, and **P.O.B. accuracy** when computing the position of one point with respect to another. Note: In surveying, P.O.B. stands for "Point of Beginning" and is accepted as the local origin for a survey or project.

Figure 2 is a diagram of the GSDM and lists the core concepts surrounded by various disciplines that, to one degree or another, use geographic information systems based upon the National Spatial Data Infrastructure (NSDI) as defined by President Clinton's Executive Order 12906 (Clinton 1994). The GSDM fully supports the NSDI and is compatible with details of that Executive Order.

Features of the Global Spatial Data Model (GSDM)

It is not possible or feasible to describe all the features of the GSDM here, but several of the more prominent features are:

- The GSDM uses standard existing mathematical equations of solid geometry and vector algebra for defining the location of a point and for manipulating spatial data. Many complicated geometrical geodesy equations, needed when working with the mathematical ellipsoid model, can be avoided. Instead, using a rotation matrix, the GSDM provides an efficient way to view any set of global X/Y/Z points in terms of local "flat earth" coordinate differences. Additionally, no geometrical integrity is lost (due to the model) and traditional coordinate systems (latitude/longitude or map projection) can still be used as desired.

- The GSDM utilizes rigorous error propagation concepts in all three dimensions and provides a standard set of tools that can be used by various disciplines to describe spatial data accuracy with statistical reliability. Metadata are still important but, in many cases, standard deviations are more efficient because they provide a numerical filter for categorizing the accuracy of each point; horizontal, vertical, or both.
- **In particular, the GSDM:**
    1. Provides a globally unique location tag for each point.
    2. Uses one set of equations world-wide. No zone constants are required.
    3. Preserves geometrical integrity both locally and globally.
    4. Is three-dimensional and does not distort distances as does a 2-D map projection. The grid/ground distance dilemma is solved.
    5. Combines horizontal and vertical into a single 3-D database.
    6. Permits each user to know the 3-D positional accuracy of each point.

Model and Perspective

The best model is one that is simultaneously simple and appropriate. Current horizontal and vertical models as used in geodesy and other spatial data applications are really not very simple but, given the traditional use of horizontal and vertical datums, have been used because they are more appropriate than the flat earth model. Burkholder (1998) describes various models used for spatial data, gives background and details of the GSDM, and includes equations for using the GSDM.

With the advent of modern data collection instruments, electronic storage media, and geographic information systems (GIS's) the demand for spatial data has mushroomed. Productivity in map-making skyrocketed because traditional manual processes were computerized and increases in productivity continue by using automated data collectors and faster computers. But the question must be asked, "Are the underlying spatial data models

really appropriate?" For the way humans visualize the world, the answer might be "yes." But for automated data collection, electronic processing, and digital storage, the answer is "no." Modern GIS's have evolved from 2-dimensional databases (state plane coordinates) to 2.5-dimensional systems where elevation is an attribute of location to pseudo 3-D systems in which state plane coordinates are used with elevations (called pseudo because elevations are referenced to an irregularly curved surface - the geoid). The next step is to use a truly 3-dimensional model that is both simple and appropriate--the GSDM which accommodates new technology, modern practice, AND permits each user to view the world from any perspective.

Perhaps user perspective is the most novel idea associated with the GSDM. The origin of a local rectangular coordinate system travels with the user and is placed (or moved) at will. The unique three-dimensional location of each point is stored in the database but, given a data set of points (10, 50, 1000 or millions), the user selects any point as the Point of Beginning (POB) and all other points are brought out of the database with respect to the chosen POB. The technology of data visualization is already in place and, using a moveable POB, it is possible to walk or fly through a proposed development using virtual reality. With the GSDM, the difference is that all points in the database represent real world points. And, the added benefit is that the GSDM provides an efficient way to describe the spatial accuracy of each point--in three dimensions.

The GSDM is already defined and in place. Using the GSDM is a matter of deciding to do so. It supports interoperability. It is seamless, simple, and rigorous. And, data quality is defined in terms of standard deviations of each component. The phrase "global spatial data model" is generic and can be freely used. The term "BURKORD" has been trademarked and applies to 1) prototype software written by the author for performing 3-D coordinate geometry and error propagation computations and 2) design of a 3-D data- base for storing geocentric X/Y/Z coordinates and associated covariances and correlations.

Example Using Points On/Near NMSU Campus

A detailed example is posted at www.zianet.com/globalcogo/example.html, but the BURKORD™ computer printout in Figure 3a and 3b shows:

1. Headings and administrative information.
2. Defining input for 2 A-order stations. Station "Reilly" is on the NMSU campus and "Crucesair" is at the airport.
3. A 3-D geocentric coordinate difference traverse (GPS base line) to two other points.
4. A listing of the 4 points as stored in the database (1 line per point). Note, format is point number, X/Y/Z coordinates, covariance values, and point name/description.
5. An expanded listing for point 1001 showing both geocentric and local covariance matrices.
6. Computed inverses between stated points which show in sequence:
    1. Standpoint geocentric coordinates, latitude/longitude/height, and local standard deviations.
    2. The computed ?X/?Y/?Z components along with their standard deviations.
    3. The computed ?e/?n/?u components along with their standard deviations.
    4. The local tangent plane distance and azimuth along with their standard deviations.
    5. Forepoint geocentric coordinates, latitude/longitude/height, and local standard deviations.
7. A listing of local tangent plane coordinate differences with respect to a user-selected P.O.B.

Applications to Current Initiatives

Many persons and organizations are doing impressive things with digital spatial data. Decentralization and the freedom to innovate is fundamental to progress, but the importance of a simple standard underlying spatial data model is the point of this paper. The GSDM is a unifying concept that provides each researcher and user the luxury of complete freedom with respect to how spatial data are used. On the other hand, complete interoperability and compatibility are assured to the extent that users are also willing to define the relationship of their database to the GSDM. In most cases, that is already being done by default if not by specific declaration.

The following list is incomplete, but intended to be somewhat representative of large-scale efforts that could benefit from adopting the GSDM as a matter of policy. It would not restrict the prerogative of any initiative, but it would insure compatibility between disciplines, between agencies, between government and industry, and between users all over the world.

1. Digital Earth Initiative http://digitalearth.gsfc.nasa.gov
2. National Integrated Lands System www.blm.gov/nils
3. XY Project www.xyproject.org
4. Global Spatial Data Infrastructure www.gsdi.org
5. The National Map http://nationalmap.usgs.gov

Figure 3a Computer Printout Showing 3-D Coordinate Geometry Computations:

```
 BURKORD(TM) COMPUTES 3-D COORDINATE GEOMETRY POSITIONS FOR SPATIAL  DATA UTILIZING
GPS VECTORS, LOCAL COORDINATE DIFFERENCES AND  3-D SURVEYING MEASUREMENTS.   COPYRIGHT
(C) 1999 AND           USE OF BURKORD(TM) LICENSED TO:  ALL RIGHTS RESERVED BY:
YOUR NAME GOES HERE                              EARL F. BURKHOLDER
COMPANY OR AGENCY NAME                           P.O. BOX 3162
ADDRESS                                        LAS CRUCES, NM 88003
CITY/STATE/ZIP/COUNTRY                       USER:   Earl F. Burkholder
DATE:   January 2, 2002       PROGRAM:        BURKORD(TM) - VERSION 8G.01, MAY 1999
S/N 8G599001  DATA FILE:      FIGPAPER.DAT  OUTPUT FILE:    FIGPAPER.OUT
CLIENT/AGENCY:  FIG Conference - Washington D.C. April 2002
JOB/PROJECT:    3-D Example Using Points on/near NMSU Campus                    * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
DEFINE X/Y/Z   1001  -1556177.6150 M  -5169235.3190 M   3387551.7090 M  Reilly HARN
"A"    COVAR MATRIX E/N/U/EN/EU/NU  .25E-04  .25E-04  .25E-04  .00E+00  .00E+00
.00E+00 METERS SQD   DEFINE X/Y/Z   1002  -1571430.6720 M  -5164782.3120 M
3387603.1880 M  Crucesair "A"      COVAR MATRIX X/Y/Z/XY/XZ/YZ  .25E-04  .25E-04
```

```
.25E-04  .00E+00  .00E+00  .00E+00 METERS SQD   FORWARD BY 3-D DX/DY/DZ  1001 TO  1003
-32.1330    -51.1750    -94.1980  Bromilow           COVAR MATRIX DX/DY/DZ/DXY/DXZ/DYZ
.10E-05  .40E-05  .10E-05  .00E+00  .00E+00  .00E+00  METERS SQD   FORWARD BY 3-D
DX/DY/DZ  1001 TO  1004    -337.8590    179.1660    104.9890  Wakeman            COVAR
MATRIX DX/DY/DZ/DXY/DXZ/DYZ   .10E-05  .10E-05  .10E-05  .00E+00  .00E+00  .00E+00
METERS SQD  A LISTING OF POINTS IN ACTIVE PROJECT IS:    1001 -1556177.6150 -
5169235.3190  3387551.7090 .000025 .000025 .000025 .000000 .000000 .000000 Reilly HARN
"A"   1002 -1571430.6720 -5164782.3120  3387603.1880 .000025 .000025 .000025 .000000
.000000 .000000 Crucesair "A"     1003 -1556209.7480 -5169286.4940  3387457.5110
.000026 .000029 .000026 .000000 .000000 .000000 Bromilow          1004 -1556515.4740 -
5169056.1530  3387656.6980 .000026 .000026 .000026 .000000 .000000 .000000 Wakeman
```

Figure 3b Computer Printout Showing 3-D Coordinate Geometry Computations:

```
 AN EXPANDED LISTING OF POINT  1001  1001     Reilly HARN "A"
X        Y        Z          E        N        U  LAT (N+S-)   32 16 55.929040 X: -
1556177.6150  X  .25E-04                  E  .25E-04  LON (E+W-) -106 45 15.160703
Y: -5169235.3190  Y  .00E+00  .25E-04          N  .00E+00  .25E-04  EL HGT
1166.5703 M Z:  3387551.7090  Z  .00E+00  .00E+00  .25E-04  U  .00E+00  .00E+00  .25E-
04  INVERSE BETWEEN POINTS   1001     Reilly HARN "A"  X = -1556177.6150  LAT (N+S-)
32 16 55.929040    +/-  .0050 METERS    N  Y = -5169235.3190  LON (E+W-) -106 45
15.160703    +/-  .0050 METERS    E   STANDARD DEVIATIONS   Z =  3387551.7090  EL HGT
1166.5703 M    +/-  .0050 METERS    U    DELTA X/Y/Z WITH SIGMAS    -337.8590M +/-
.007M    179.1660M +/-  .007M    104.9890M +/-  .007M   DELTA E/N/U WITH SIGMAS    -
375.1645M +/-  .007M    128.3724M +/-  .007M     -6.6293M +/-  .007M   LOCAL PLANE
INV: DIST =     396.5197M +/-  .007M  N AZI. = 288 53 23.04 +/-   3.7 SEC  1004
Wakeman          X = -1556515.4740  LAT (N+S-)   32 17   .095547    +/-  .0051 METERS
N  Y = -5169056.1530  LON (E+W-) -106 45 29.495387    +/-  .0051 METERS    E
STANDARD DEVIATIONS   Z =  3387656.6980  EL HGT          1159.9533 M    +/-  .0051
METERS    U    DELTA X/Y/Z WITH SIGMAS     305.7260M +/-  .007M   -230.3410M +/-
.007M   -199.1870M +/-  .007M   DELTA E/N/U WITH SIGMAS     359.1568M +/-  .007M   -
239.1158M +/-  .007M      5.5524M +/-  .007M   LOCAL PLANE INV: DIST =     431.4742M
+/-  .007M  N AZI. = 123 39 16.14 +/-   3.5 SEC  1003     Bromilow         X = -
1556209.7480  LAT (N+S-)   32 16 52.334086    +/-  .0052 METERS    N  Y = -
5169286.4940  LON (E+W-) -106 45 15.772679    +/-  .0051 METERS    E   STANDARD
DEVIATIONS   Z =  3387457.5110  EL HGT          1165.5203 M    +/-  .0053 METERS    U
LISTING OF POINTS WITH RESPECT TO MASTER P.O.B:          1001  Reilly HARN "A"
(ASSUMING POSITION OF P.O.B. IS ERRORLESS)    NUMBER      NORTH  SIGMA     EAST  SIGMA
UP  SIGMA    STATION     1002    -27.496  .005 -15889.221  .005    139.911  .005
Crucesair "A"       1003   -110.757  .005    -16.017  .005     -1.051  .005
Bromilow            1004    128.372  .005   -375.165  .005     -6.629  .005    Wakeman
```

References

Burkholder, E.F., 1999, "Spatial Data Accuracy as Defined by the GSDM," Surveying & Land Information Systems, Vol. 59, No. 1, pp 26-30. www.zianet.com/globalcogo/accuracy.pdf

Burkholder, E.F., 1998, "A Practical Global Spatial Data Model for the 21st Century," presented at the National Technical Meeting of the Institute of Navigation, Long Beach, CA, January, 1998. www.zianet.com/globalcogo/ionpaper.pdf

Burkholder, E.F., 1997, "Definition and Description of a Global Spatial Data Model," U.S. Copyright Office, Washington D.C. www.zianet.com/globalcogo/gsdmdefn.pdf

Clinton, W.J. (President), 1994; "Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure," Executive Order 12906, signed April 11, 1994, Federal Register, vol. 59, No. 71, pp 17671-17674.
www.fgdc.gov/publications/documents/geninfo/execord.html

DMA (Defense Mapping Agency), 1991, "Department of Defense World Geodetic System 1984: Its Definition and Relationships With Local Geodetic Systems," Technical Report 8350.2, Defense Mapping Agency (now National Imagery and Mapping Agency), Fairfax, Virginia.

Gore, Al, (Vice President), 1998, "The Digital Earth: Understanding our planet in the 21st century," www.digitalearth.gov/speech.html speech purportedly, but not actually, given at the California Science Center, Los Angeles, California, January 31, 1998.

Leick, Alfred, 1995, "GPS Satellite Surveying," John Wiley & Sons, NY, NY.

Mikhail, Edward, 1976; "Observations and Least Squares," Harper & Row, New York, NY.

Wolf, P.R. and C.D. Ghilani, 1997, "Adjustment Computations: Statistics and Least Squares in Surveying and GIS," John Wiley & Sons, NY, NY.

Chapter 5:

EASE-Grid:
A Versatile Set of Equal-Area Projections and Grids

by *Mary J. Brodzik and Kenneth W. Knowles*

National Snow & Ice Data Center
Cooperative Institute for Research in Environmental Science
University of Colorado, Boulder, CO, USA

Brodzik, M. J., & Knowles, K. (2002). EASE-Grid: a versatile set of equal-area projections and grids. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis.

Abstract

The National Snow and Ice Data Center's (NSIDC) Equal-Area Scalable Earth Grid (EASE-Grid) comprises three equal-area projections, combined with an infinite number of possible grid definitions. The EASE-Grid is based on a philosophy of digital mapping and gridding definitions developed at the University of Colorado at Boulder.

The original, 25 kilometer grids were defined for data products generated by the NOAA/NASA Level 3 Passive Microwave Pathfinder Projects, which include gridded passive microwave brightness temperatures and related geophysical products derived from the brightness temperatures. However, the grids have been adopted by a number of other projects, including the TIROS Operational Vertical Sounder (TOVS) and Advanced Very High Resolution Radiometer (AVHRR) Polar Pathfinders, the Arctic and Antarctic Research Institute, St. Petersburg, Russia (AARI) Sea Ice data, and NSIDC's EASE-Grid versions of the Global Land Cover Classification (GLCC) data and the International Permafrost Association Permafrost and Ground Ice Map.

The gridding philosophy used to implement a library of software routines is based on the assumption that a gridded data set is completely defined by two abstractions, the map projection and an overlaid lattice of grid points, often referred to as "cells." The software may be used to define all standard projections, and convert gridded data between them. However,

this presentation will be restricted to an overview of the family of specific projections and grids that we have called the NSIDC EASE-Grid, or simply EASE-Grid. The discussion will include how to define a map projection and grid definition in general, and our reasons for choosing equal-area projections and a spherical Earth model for our target applications in particular. It will include some examples of applications and an intercomparison between EASE-Grid data sets.

Introduction

The Equal-Area Scalable Earth Grid (EASE-Grid) comprises three equal-area projections, combined with an infinite number of possible grid definitions. It is based on a philosophy of digital mapping and gridding definitions that was developed at the National Snow and Ice Data Center (NSIDC), in Boulder, CO. This philosophy was used to implement a library of software routines, based on the assumption that a gridded data set is completely defined by two abstractions, the map projection and an overlaid lattice of grid points. The complete source code is available via ftp, and contains software to convert among many projections, but this paper will be restricted to an overview of the family of specific projections and grids that we have called the NSIDC EASE-Grid, or simply EASE-Grid.

The EASE-Grid is intended to be a versatile scheme for users of global-scale gridded data, specifically remotely sensed data, although it is gaining popularity as a common gridding format for data from other sources as well. We begin with a short introduction to the abstractions used in NSIDC's generic mapping and gridding software, and proceed to the specific projections and grids that comprise the EASE-Grid family of grids. We include several short descriptions of the EASE-Grid used for different projects, as examples of the flexibility of the format. We conclude with an example of two EASE-Grid data sets that are used to study sea ice concentrations in Baffin Bay.

Defining a Gridded Data Set

NSIDC's mapping software is based on the assumption that a gridded data set is completely defined by the map projection and an overlaid lattice of grid points, often referred to as "cells". It is useful to think of the projection and the grid lattice as separate but related abstractions. The projection is simply a mathematical coordinate transformation of points on the curved surface of the Earth to points on a plane. The lattice of grid points can be imagined as a transparent piece of graph paper, overlaid on the plane of projection and anchored to it at a specified point.

Once the projection is chosen, any number of grid definitions can be used to describe the effect of changing the grid, or "graph paper," for the application at hand. For example, smaller graph cells can be used when a higher resolution is needed, or the size of the graph paper (number of columns/rows) can be reduced to study a subset area of the full projection.

An array of gridded data, then, consists of one data element for each grid "cell" or lattice point. The user has complete flexibility to define the meaning of grid cell values, according to the most appropriate sampling technique for the data and application at hand. In many cases, particularly with remotely sensed imagery, it is important for a user to think of gridded data elements as values associated with the lattice points of the graph paper, rather than as associated with the "area of the grid cell." We have found the more general lattice point concept to be more useful when the data represent regularly sampled measurements in a continuous field, for example, passive microwave brightness temperatures, or visible wavelength radiances. In the sensor swath space, the data value is simply a sample measurement from the continuous field, and may or may not have any physical relationship to the size and shape of the eventual grid "cell" surrounding the eventual lattice point in the regularly gridded data array. For example, although the sampling interval of a 19 GHz scanning passive microwave radiometer might be 25 km, the effective field of view of the antenna might be an elliptical area, 40 km x 60 km. The eventual gridded data array might be defined on a 25 km grid, with data element values chosen from the nearest neighbor of the latest swath. In this case, the data element still represents a 40 km x 60 km brightness temperature.

It is entirely up to the data set producer to define the meaning of a gridded data element during the sampling process. The process of sampling data to the regular grid is sometimes referred to as binning. Examples of binning techniques include averaging all data that falls into a given cell, or taking a maximum, minimum, median or latest value in the cell. The producer might choose to involve the area and shape of the cell in the definition of a data element, but is certainly not required to do so. Alternative methods include nearest neighbor, bilinear interpolation, or otherwise weighted averages of surrounding data samples.

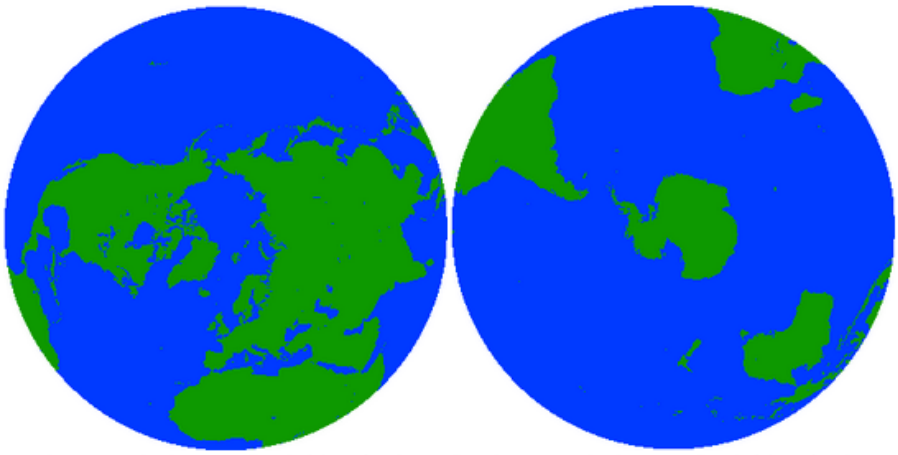
EASE-Grid Map Parameters

The EASE-Grid was originally developed at NSIDC for the data products generated by the NOAA/NASA Pathfinder Program Special Sensor Microwave Imager (SSM/I) project, which included gridded passive microwave brightness temperature and related geophysical products derived from the brightness temperatures at a relatively coarse 25 km resolution (Armstrong and Brodzik, 1995; Armstrong, Brodzik and Varani, 1997). However, given the flexibility of an infinite number of grid definitions for the EASE-Grid projections, the format has since been adopted by a number of other projects, with grid resolutions ranging from 1.25 km to 250 km. In addition to the original SSM/I Pathfinder data, these include:

- Advanced Very High Resolution Radiometer (AVHRR) Polar Pathfinder Twice-Daily 1.25 km EASE-Grid Composites
- Arctic and Antarctic Research Institute (AARI) 10-Day Arctic Ocean EASE-Grid Sea Ice Observations
- Arctic Climatology Project - EWG Arctic Meteorology and Climate Atlas
- Nimbus-7 Scanning Multichannel Microwave Radiometer (SMMR) Pathfinder Daily EASE-Grid Brightness Temperatures
- TIROS-N Operational Vertical Sounder (TOVS) Pathfinder Path-P Daily Arctic Gridded Atmospheric Parameters

Ancillary EASE-Grid data sets include NSIDC's EASE-Grid versions of the Global Land Cover Classification (GLCC) data and the International Permafrost Association Permafrost and Ground Ice Map. Instrument teams for the recently launched (December, 1999) Moderate Resolution Imaging Spectroradiometer (MODIS) and the upcoming Advanced Microwave Scanning Radiometer (AMSR) sensors plan to release gridded snow and ice products in EASE-Grid.

The EASE-Grid projections comprise polar aspect spherical Lambert azimuthal equal-area projections (Snyder, 1987, p. 186), for the Northern or Southern hemisphere, respectively, and a "modified" cylindrical equal-area projection (Snyder, 1987, p. 77; Maling, 1992, p. 431) for applications in mid- and low-latitude regions.



Northern Hemisphere EASE-Grid projection Southern Hemisphere EASE-Grid projection

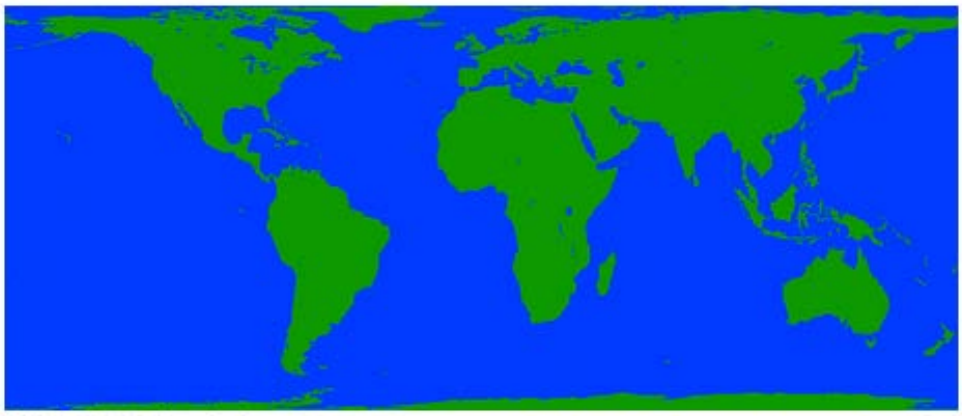
The North azimuthal equal-area map is defined as

- r = (ncols - 1) - r_0 + 2R/C sin(***pi***/4 - ***p***/2) sin(***l***)
- s = (nrows - 1) - s_0 + 2R/C sin(***pi***/4 - ***p***/2) cos(***l***)
- k = sec(***pi***/4 - ***p***/2)

The South azimuthal equal-area map is defined as

- r = (ncols - 1) - r_0 + 2R/C cos(***pi***/4 - ***p***/2) sin(***l***)
- s = (nrows - 1) - s_0 - 2R/C cos(***pi***/4 - ***p***/2) cos(***l***)

- k = csc(***pi***/4 - ***p***/2)



Global EASE-Grid projection

The modified cylindrical equal-area map is defined as

- r = (ncols - 1) - r_0 + R/C ***l*** cos(p_s)
- s = (nrows - 1) - s_0 - R/C sin(***p***) / cos(p_s)
- k = cos(p_s) / cos(***p***)

where:

- k = particular scale along parallels
- h = 1/k (particular scale along meridians)
- ***l*** = longitude in radians
- ***p*** = latitude in radians
- R = radius of the Earth = 6371.228 km
- C = nominal cell size (km)
- ncols = number of grid columns
- nrows = number of grid rows
- r = column coordinate (increasing to the right)
- s = row coordinate (increasing from top to bottom)
- r_0 = map origin column

- s_0 = map origin row
- p_s = ***pi***/6 radians = 30 degrees (cylindrical projection standard parallel)

Both projections are based on a spherical model of the Earth with radius R = 6371.228 km. This radius was chosen for historical reasons.

The values of C, ncols, nrows, r_0 and s_0 are determined by the grid that is chosen to overlay the projection.

Why "equal-area" maps?

We chose equal-area projections over other possibilities for the EASE-Grid, because our original application was a fixed geographic look-up table for storage and retrieval of satellite passive microwave brightness temperatures. We were using an optimal interpolation binning method to derive brightness temperatures at fixed grid locations, effectively yielding what the sensor would have seen if it had been pointed at the center of the fixed cell. Therefore, a rectangular grid lattice superimposed on an equal-area map most faithfully represented the nominal passive microwave footprint.

On equal-area maps, a small circle placed anywhere on the map will always cover the same amount of area on the globe, and, at any point on the map, the product of the scale h along a meridian of longitude and the scale k along a parallel of latitude is always one, that is, h = 1/k. The aspect ratio, $k/h = k^2 = 1/h^2$, and angular deformation, $\omega = 2 \arcsin(|h - k|/(h + k))$, are measures of shape distortion.

For the Northern and Southern hemisphere EASE-Grid projections, the aspect ratio varies from 1 at the pole to 1.17 at 45N and increases to only 2 at the equator. For the global EASE-Grid projection, the aspect ratio varies more widely (see details in the following table). The selection of +/-30 degrees for the standard parallels of the cylindrical projection gives a map with minimum mean angular distortion over the continents. This projection is intended for the study of parameters in the mid- to low-latitudes.

Aspect ratios and angular deformation (measures of shape distortion) of the EASE-Grid projections:

Azimuthal Equal-Area			Cylindrical Equal-Area		
latitude	k/h	w	latitude	k/h	w
90°	1.00	0°	80°	24.90	134°
75°	1.02	1°	75°	11.20	113°
60°	1.07	4°	60°	3.00	60°
45°	1.17	9°	45°	1.50	23°
30°	1.33	16°	30°	1.00	0°
15°	1.59	26°	15°	0.80	12°
0°	2.00	39°	0°	0.75	16°

In contrast, on conformal maps, shapes within a small area are reproduced accurately, so a small circle on the globe will look like a small circle on the map. At any point on the map, the scale h along a meridian of longitude is equal to the scale k along a parallel of latitude, and kh - 1 is a measure of areal distortion. For example, NSIDC produces other polar gridded data products using a polar stereographic map true at 70N. The projection is a conformal map. By definition, the aspect ratio remains 1 everywhere, however, the areal distortion of this map varies from -6% at the pole to +29% at 45N and increases to +276% at the equator. Angular deformation varies from 0° at 70N/S to 4° at the pole and 71° at the equator.

Areal distortion and angular deformation of the Polar Stereographic map true at 70N:

Polar Stereographic, (true at 70N)		
latitude	kh - 1	w
90°	-6%	4°
45°	29%	15°
0°	276%	71°

A very popular map that is neither equal-area nor conformal is the cylindrical equidistant map, also known as the "lat-lon grid." This map suffers from both areal and shape distortion, and angular deformation, as follows:

Cylindrical Equidistant			
	Shape Distortion	Areal Distortion	Angular Deformation
latitude	k/h	kh - 1	*w*
89°	57	5630%	176°
80°	6	476%	141°
60°	2	100%	74°
45°	1.4	41%	39°
0°	1	0%	0°

In summary, given the choices of either shape distortion or areal distortion or both, we decided in favor of the equal-area projections for the EASE-Grid because they minimized the amount of distortion over the hemispheric and global scale we were attempting to portray. One convenient side effect of this choice is that calculations of areal statistics are reduced to simply summing pixels and multiplying by a constant area per pixel, so the acronym, "EASE-" takes on a secondary meaning, as in "easy to use."

Why a Spherical Earth Model?

Another question that is sometimes raised is why we chose to use a spherical earth model over an elliptical model, and how much "error" this introduces in the gridding geolocation. Keeping in mind that the EASE-Grid is simply a geographic data storage and retrieval mechanism, the answer is that no error is introduced by this model choice.

Representation of the gridded data as a fixed array of values is accomplished with a set of equations to map from geographic coordinates (latitude, longitude) to grid coordinates (column, row). In this sense, the location (column and row) of each grid "cell" can just be considered an entry in a look-up table, i.e. a place to store the data (brightness temperature, albedo, time stamp, etc.) for a specific, implicitly defined, geographic location. As long as the transformation back from grid coordinates (column,row) to geographic coordinates (latitude, longitude) is performed with the inverse transformation that uses the same Earth model, there is no error introduced by using a spherical Earth model. Choice of an elliptical model would only

slow down the transformation calculations, (geographic to grid and back), with no gain in accuracy.

EASE-Grid Family of Grid Definitions

A grid is always defined in relation to a specific map projection. It is essentially the parameters necessary to define a rectangular coordinate system overlaid on a flat map and anchored to it at the map origin. The following four elements completely describe a grid:

- the map projection
- the numbers of columns and rows
- the number of grid cells per map unit (the map unit is part of the projection parameters)
- the grid cell coordinates of the map's origin

The EASE-Grid family of grid definitions includes, but is not limited to, the following specific grids.

The Original SSM/I Grids

The original 25 kilometer grids were defined for the data products generated by the SSM/I Level 3 Pathfinder Project at NSIDC. The sampling resolutions of the SSM/I brightness temperature data were 25 km and 12.5 km. The original grids were designated "low" (25 km) and "high" (12.5 km) resolution, although these are, of course, relative terms. The "low" resolution grids have a nominal cell size of 25 km x 25 km. A slightly larger actual cell size C=25.067525 km was chosen to make the full global, 25 km ("Ml", the uppercase "M" stands for "Modified" cylindrical equal-area, or "Mid-"latitude; the lowercase "l" stands for "low" resolution) grid exactly span the equator, and was then used for all three projections for the sake of data product consistency. Of course, few cells actually have these dimensions, but they all have the same area.

By convention, grid coordinates (r,s) start in the upper left corner, at cell (0,0), with r increasing to the right and s increasing downward (Knowles, 1993). Rounding the grid coordinates up at .5 yields the grid cell number. A grid cell is centered at grid coordinates (j,i) and bounded by: (j - .5) <= r < (j + .5) and (i - .5) <= s < (i + .5).

The 25 km hemispheric grids for the polar aspect azimuthal projections (aka "Nl" for "Northern Hemisphere, low resolution" and "Sl" for "Southern Hemisphere, low resolution") are defined with 721 columns, 721 rows, with the respective pole anchored at cell (360.0,360.0). The Ml grid for the cylindrical projection is defined with 1383 columns, 586 rows, with the point where the equator crosses the prime meridian at cell location (691.0,292.5).

For each 25 km grid, the set of corresponding "high" resolution (12.5 km) grids, "Nh", "Sh" and "Mh", was defined such that the grid coordinates are coincident (aka "bore-centered") and exactly double the lower resolution grid coordinates. The Ml grid is symmetrical about the prime meridian, but the Mh grid is not. The (25 km) Ml grid exactly spans the equator, from 180 W to 180 E, with 1383 grid cells. The (12.5 km) Mh grid, also exactly spans the equator, with 2766 grid cells. However, since the center of the Ml column 0 is coincident with the Mh column 0, the western edge of the Mh grid cell in column 0 row 293 (at the equator) is slightly east of 180 W, and the eastern edge of the Mh grid cell in column 2765 row 293 is slightly east of 180 E.

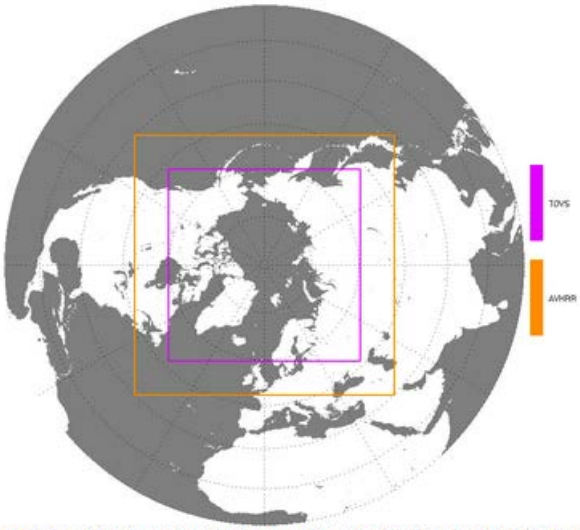
The dimensions, center, and extent of the original SSM/I grids are summarized below. It is important to remember that there is nothing specific to the SSM/I data in these definitions. If these grid definitions are considered appropriate for another data set, they can be used with no changes.

Original "Low" (25 km) and "High" (12.5 km) Resolution SSM/I Grids										
Grid	Dimensions		Map Origin		Map Origin		Grid Extent			
Name	Width	Height	Column (r_0)	Row (s_0)	Latitude	Longitude	Minimum Latitude	Maximum Latitude	Minimum Longitude	Maximum Longitude
Ml	1383	586	691.0	292.5	0.0	0.0	86.72S	86.72N	180.00W	180.00E
Mh	2766	1171	1382.0	585.0	0.0	0.0	85.95S	85.95N	179.93W	180.07E
Nl	721	721	360.0	360.0	90.0N	0.0	0.34S	90.00N	180.00W	180.00E
Nh	1441	1441	720.0	720.0	90.0N	0.0	0.26S	90.00N	180.00W	180.00E
Sl	721	721	360.0	360.0	90.0S	0.0	90.00S	0.34N	180.00W	180.00E
Sh	1441	1441	720.0	720.0	90.0S	0.0	90.00S	0.26N	180.00W	180.00E

Other Grid Definitions in the EASE-Grid Family

The Polar Pathfinders

Users of the NSIDC EASE-Grid are not limited to the grid orientation, size and resolution described above, and are free to define grids that are more appropriate for a given data set. For example, the TOVS Polar Pathfinder data were defined with the EASE-Grid Northern hemisphere map projection parameters, and a polar subset of the original hemisphere at a 100 kilometer resolution. The AVHRR Polar Pathfinder data were defined for both Northern and Southern hemisphere maps, as subsets of each, at 1.25 km, 5 km, and 25 km resolutions. The figure below shows the Northern hemisphere grid extent for SSM/I (the full hemisphere), TOVS Polar, and AVHRR Polar grids (respective subsets).



Relative Northern hemisphere grid extents of Polar Pathfinders (SSM/I (full hemisphere), AVHRR and TOVS).

The AARI Sea Ice Data in EASE-Grid

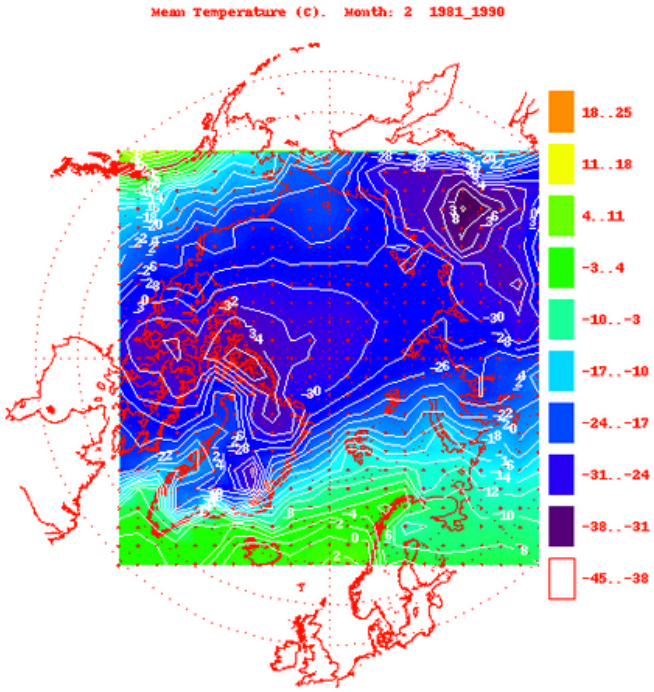
The Arctic and Antarctic Research Institute (AARI) 10-Day Arctic Ocean EASE-Grid Sea Ice Observations data provide another example. These data did not require hemispheric coverage, but the data set producers at NSIDC wanted to provide them in a grid that would facilitate intercomparison with sea ice data derived from SSM/I. Therefore the AARI EASE-Grid was defined to be the subset of the SSM/I Pathfinder Nh grid (Northern hemisphere, 12.5 km resolution) defined by columns 360 through 1080 and rows 360 through 1080. The resulting AARI EASE-Grid is 721 columns and 721 rows. This, in turn, relates the AARI EASE-Grid definition to the 25 km AVHRR EASE-grid (aka "NA25") subset via the following simple relationship:

- AARIcolumn = 2 * NA25column
- AARIrow = 2 * NA25row

The Arctic Climatology Project Arctic Meteorology and Climate Atlas in EASE-Grid

NSIDC has produced an atlas of Arctic meteorology and climatology under the auspices of the U.S.-Russian Joint Commission on Economic and Technological Cooperation's Environmental Working Group (EWG). The gridded fields produced for this atlas are defined for a subset of the

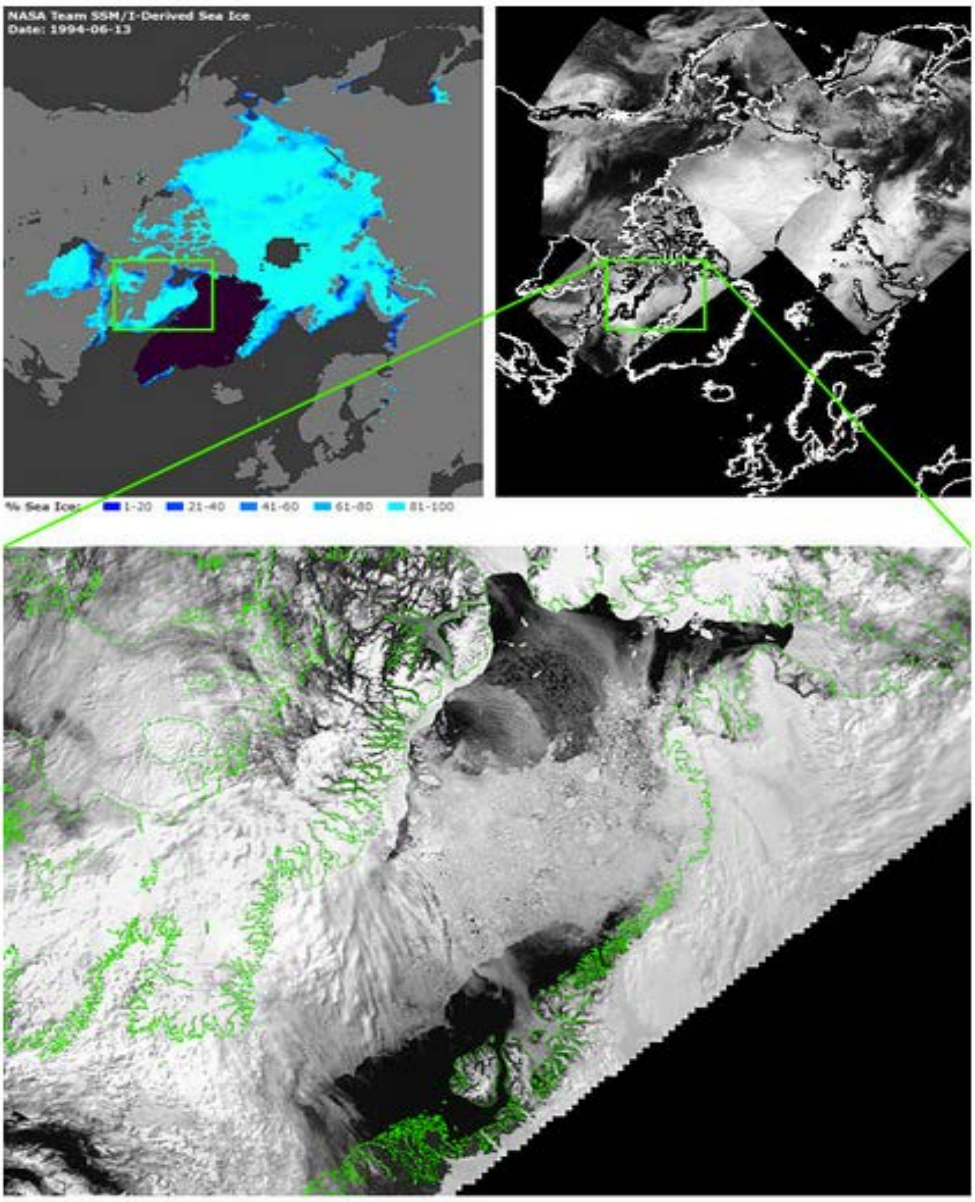
full northern hemisphere above 65N, at a 250 km grid spacing. Data arrays are 23 columns by 23 rows. An example temperature field is included below.



Mean air temperature (degrees Centigrade), February, 1981 - 1990.
(Dots indicate the EWG EASE-Grid lattice points.)

An Example Application

The following set of images illustrates a comparison of SSM/I-derived (top left) sea ice in Baffin Bay (the water body to the west of Greenland) with AVHRR-derived albedo (top right and zoomed area, below) for June 13, 1994. Since all images are derived from various resolution grids of the same NI EASE-Grid projection, the reader can easily visually compare the ice edge. Digital comparison and analysis are likewise easily performed using the user's favorite graphical analysis package.



Arctic Satellite-derived Sea Ice, June 13, 1994. Top left image represents SSM/I-derived (passive microwave) sea ice concentrations at 25 km resolution; top right image represents 25 km AVHRR Albedo Browse Product; bottom image represents zoomed area of Baffin Bay, AVHRR Albedo Product full resolution (1.25 km).

Conclusions

While originally intended for use with a single data product, the EASE-Grid has proven to be flexible and extensible to other global, gridded applications. The projection and gridding abstractions are simple and easy to apply to the requirements of a new data set. Data from diverse sources can be resampled and expressed as digital arrays of varying resolutions, which

are defined in relation to one of three possible projections. Storage as a simple digital array facilitates portability and ability to be imported into a user's favorite analysis package. Users find that visualization and intercomparison operations are then greatly simplified, and that the tasks of analysis and intercomparison can be more readily accomplished.

References and Useful Web Sites

- Armstrong, R. L. and M. J. Brodzik. 1995. An Earth-Gridded SSM/I Data Set for Cryospheric Studies and Global Change Monitoring. Adv. Space Res. 16(10):155-163.
- Armstrong, Richard, M. J. Brodzik, A. Varani, 1997. "The NSIDC EASE-Grid: Addressing the need for a common, flexible, mapping and gridding scheme." Earth System Monitor, Vol. 7, No. 4, 3 pp.
- Knowles, Kenneth W. 1993. Points, pixels, grids, and cells -- a mapping and gridding primer.<http://cires.colorado.edu/~knowlesk/ppgc.html>
- Maling, D. H. 1992. Coordinate Systems and Map Projections. 2nd ed. Oxford: Pergamon Press.
- National Snow and Ice Data Center. All About EASE-Grid. <http://nsidc.org/data/ease/>
- National Snow and Ice Data Center. Complete source code for mapping and gridding software.<ftp://baikal.colorado.edu/pub/NSIDC/maps.tar.gz>
- Snyder, John P. 1987. Map Projections: A Working Manual. USGS Professional Paper 1395. Washington, DC: US Government Printing Office.

Chapter 6:

Ellipsoidal Area Computations of Large Terrestrial Objects

By Hrvoje Lukatela, Geodyssey Limited

Paper presented at the International Conference on Discrete Global Grids,

Santa Barbara, 26 - 28 March 2000

Lukatela, H. (2000). Ellipsoidal Area Computations of Large Terrestrial Objects. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis. http://www.ncgia.ucsb.edu/globalgrids-book/eac/

Abstract

The mathematics of area computation on the ellipsoidal planetary surface is straightforward; it is however rarely implemented in its rigorous form. Most spatial information systems dealing with two dimensional objects treat the area not as a simple derivative of the object definition geometry, but rather as an artifact of its representation in a particular planar representation. This approach fails when no single, canonical planar representation is either practical or desirable, or when the spatial extent of the object exceeds the useful coverage of a single planar projection.

The Hipparchus geopositioning model represents two dimensional terrestrial objects in the context of an irregular global grid consisting of spheroidal Voronoi polygons. This paper presents the strategy and outlines the implementation of area computation for such objects. It assumes that efficiency is as important as is the precision, and that the objects can be of any size, shape and topological complexity. The speed and accuracy of the computation is examined by applying it to a large, global object of high data volume and considerable topological complexity.

Introduction

his paper presents a particular numerical solution to a rather straightforward and well-defined mathematical problem: given a two-dimensional object of an arbitrary complex topology on a surface of an ellipsoid of rotation, find its area. Several elements of the underlining mathematics are worth noting at the outset:

- The total surface of an ellipsoid of rotation can be expressed as a closed (but transcendental) function of its semi-minor and semi-major axes.
- The area of a pseudo-trapezoidal figure bounded by two meridians and two parallels leads to a single ellipsoidal integral over the latitude domain.
- The area of a triangle - with geodesics as its sides - leads to a double ellipsoidal integral over the geodesic lengths.

Even the mathematically simpler - trapezoidal - decomposition results in an integral that requires a binomial series expansion. In addition, the substitution of the continuously changing width of latitude-bands for a figure bounded by a geodesic leads to an approximation, the magnitude of which depends on both the band height and the local azimuth of the geodesic. The decomposition into triangles results in a similar, but even more complicated numerical solution. In either case, the decomposition of topologically "deep" two dimensional objects into either pseudo-trapezoidal or triangular components can lead to rather complex implementation problems. A good overview of the ellipsoidal geometry and the algebra and calculus used to implement common geodetic computations can be found in either Jordan (1958) or Bomford (1975).

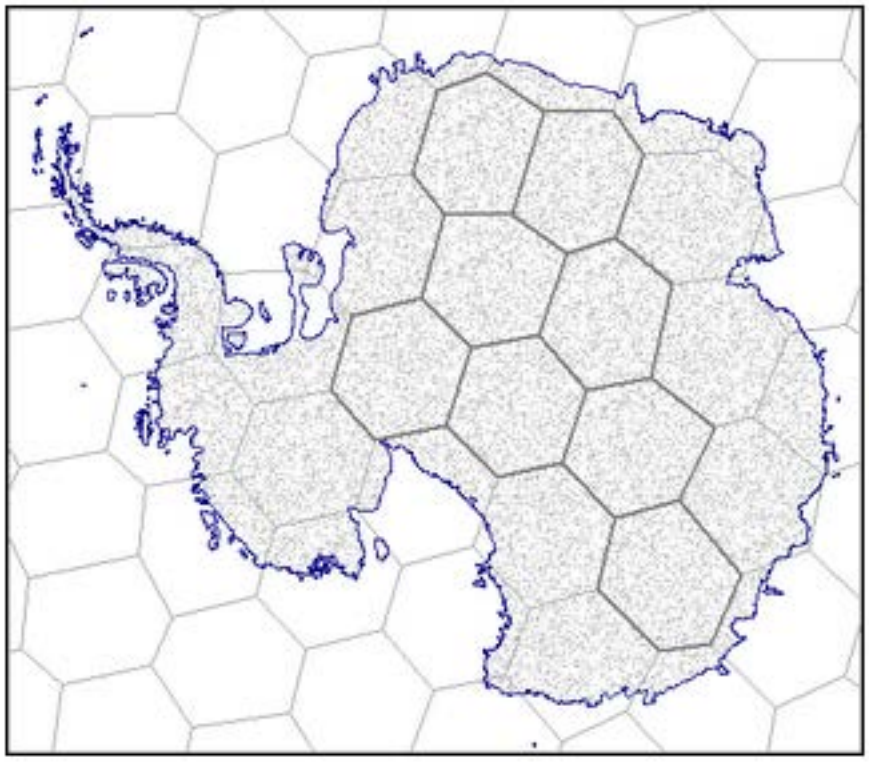


Figure 1: A two-dimensional object in a spheroidal Voronoi grid

If the two-dimensional ellipsoidal surface object is already represented in a form which provides a convenient basis for the geometrical decomposition, it will be advantageous to use such representation as a basis for the area computation. This is the case in the Hipparchus Geopositioning Model, where an object is represented in an irregular global grid of Voronoi polygons. (The full presentation of this model is outlined in a web-resident publication titled: Hipparchus Geopositioning Model: an Overview). It is an example of a "constructive index", where the object geometrical integrity remains intact, and the infrastructure required to access parts of the object that belong to one index block (data cell, tile... etc.) exists "in addition" to the data defining its geometry. Figure 1 depicts an object in a global Voronoi grid. We will - later on in this text - recapitulate those features of this type of object representation which influence directly the area computation. For now, we note only that there are some cells ("boundary cells") through which the object boundary "meanders" and some cells ("interior cells") which are completely encompassed by the object. The area of an object will be

determined by summing up the area of all interior cells and adding to this the partial area - that portion covered by the object - of all of its boundary cells.

Area of Ellipsoidal Voronoi Cells

We will first outline the method for determining the area of any Voronoi cell. This is done by subdividing the cell into triangles with neighbor edges as a base and cell center as the opposite vertex. The edges of these triangles are segments of great circles on a sphere related to the ellipsoid (on which both the object boundary and cell-center coordinates are defined by their geodetic coordinates) by conformal ellipsoid to sphere mapping. While these edges are not ellipsoidal geodesics, the maximum line displacement is both relatively small (2.5 meters for a line similar to the one discussed below in the context of UTM mapping) and the "gain" and "loss" tend to be of the same magnitude when all the edges along the cell boundary are considered. The area of these triangles is then computed by spherical productions, but using a constantly changing radius determined by the mean curvature of the ellipsoid surface in the cell center.

The method presented here shares two salient principles with the one devised by Tobler and Kimerling, as described in Kimerling (1984): the first is the use of local ellipsoidal radii in spherical triangle area productions and the second is the tracking of positive and negative area contributions based on the direction of the boundary segment. The latter parallels the procedural makeup of the planar polygon area computation and provides an effective way to keep the non-numerical programming complication within reason, while at the same time avoiding any restriction on the level of the topological complexity of the objects for which the area is calculated.

In geodetic computations, the expression for the mean radius of the ellipsoid surface in a given point is commonly referred to as Gauss' formula: r = sqrt(m*n) (where mand n are radii of the meridian and prime vertical, respectively). The conventional computation of the mean radius uses the ellipsoidal eccentricity term - an unnecessary complication in computation on digital

computers. The preferred method uses tsq: a latitude (phi) dependent square of the free term of the tangential plane, defined directly as a function of the sine and cosine of the point of tangency and the major and minor ellipsoid semi-axes (a and b respectively), as follows:

```
s = sin(phi)
c = cos(phi)
tsq = a*a*c*c + b*b*s*s
r = (a*a*b)/tsq
```

This method of mean ellipsoid radius computation is especially convenient for systems such as Hipparchus, which represent the location of a point using the direction cosines of its ellipsoid normal: the evaluations of s and c require no expensive transcendental computations. Also, the tsq term finds a repeated use in many different ellipsoid geometry propositions.

No *a priori* error term has been derived for this "finite element" ellipsoid-to-sphere approximation; the total area of all cells so computed can however be compared (and adjusted) to the total ellipsoid surface. (See below, under "Some Numerical Results", for details). Since we assume that a system will be required to produce the area of many different objects, represented in the context of the same global grid, an array of areas - one element for each cell - will be conveniently stored with the other data in the structures representing the Voronoi polygons.

Computing the area of a spherical triangle for which the lengths of all three sides (**a**, **b**and **c**) are given can be done using two different approaches: by applying Legendre's approximation, or by L'Huilere's theorem. The former states that the area of a spherical triangle will be getting closer to the area of a planar triangle with the same side lengths, as the ratio of the triangle perimeter divided by the spherical radius gets smaller. The latter is, on the other hand, a rigorous evaluation of the spherical triangle area in terms of the three sides. To compare the two:

```
P = sqrt(s*(s-a)*(s-b)*(s-c))
```

versus:

P = 4*atan(sqrt(tan(s/2)*tan((s-a)/2)*tan((s-b)/2)*tan((s-c)/2)))

(where s is the common shorthand notation for the half-sum of all three sides)

Implementations which impose a low maximum cell radius limit might take advantage of a considerably faster Legendre's approximation. The implementation used to derive the numerical examples given below is, however, based on the second (L'Huilere's) expression, applied to a "local" sphere with the radius equal to the mean ellipsoid radius at the cell center.

Area Computation in Boundary Cells

The determination of the area of an object in a boundary cell requires no additional mathematics. If an object is topologically well-defined, its area inside a boundary cell will consist of a finite number of distinct "faces", each bounded by a closed ring. The ring may consist of either the segments that all belong to the object boundary, or, in the general case, a combination of object boundary and cell edge segments. In a procedure directly paralleling the usual implementation of planar polygon area computation, we can traverse the ring, and accumulate triangular areas subtended from each segment and the coordinate origin - adding or subtracting, depending on the radial direction of the boundary segment respective to the coordinate origin.

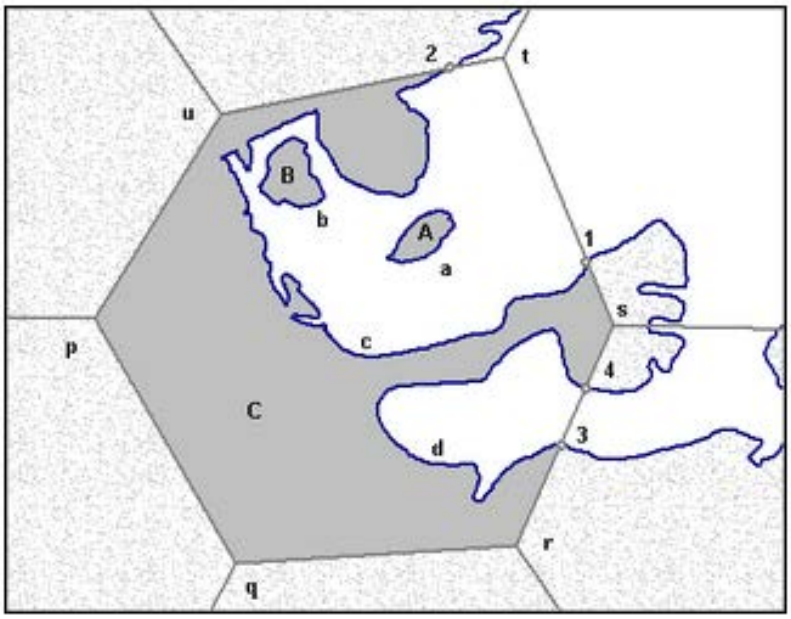


Figure 2: Geometry of a boundary cell

We will next identify some area computation pertinent features of the object geometry definition in an example of a boundary cell, as depicted in Figure 2. (It is an enlarged part of the object shown in Figure 1). The topological decomposition of the object presented in the following is essentially the same as it would be if the object was a planar one - thus additional details, definitions and code segments can be found in many sources describing planar computational geometry - for instance Bowyer (1983).

The fundamental geometrical element is a "fragment": an ordered array of vertices forming the object boundary, in either a closed ring, or starting and ending on the cell boundary. There are four boundary fragments in the example: **a**, **b**, **c** and **d**; two of them are closed rings (**a** and **b**), and two are connecting cell boundary crossing points (fragment **c**, connecting crossing points **1** and **2**; and fragment **d**, connecting points **3**and **4**). For each fragment, we note that the start point is the point where the object boundary "enters" the cell, and the fragment end point is the point where the boundary "exits" the cell.

Fragments are directed in the mathematically positive sense, such that the object interior is always on the left-hand side of the boundary line. This ensures that the "voids" will produce a

negative contribution to the total area, regardless of how "deeply nested" the topology of the object is.

One or more fragments form the boundary of a "face" - a single continuous, connected area. In the example, there are three faces: **A**, **B** and **C**. Faces **A** and **B** are formed by single ring fragments **a** and **b**. Face **C** is formed by fragments **c** and **d**. In addition to the two fragments, the boundary of face **C** also requires two cell-edge line segments: first one from the end point of the fragment **d**, to the cell vertex **s**, to the start point of the fragment **c** and the second, from the end point of fragment **c**, through the cell vertices **u**, **p**, **q** and **r**, to the start point of the fragment **d**.

A canonical representation of the Hipparchus system two-dimensional objects identifies only the distinct boundary fragments, and not the faces they form. No spatial algebra proposition (e.g., "polygon overlay") requires this knowledge, and all such propositions would thus be burdened with the additional code if required to keep track of the fragment/face relationship. The area computation algorithm must therefore "construct" the faces as and when required. This information is used implicitly, in the ring traverse order, and is not stored permanently. This "construction" is trivial for faces which are formed by a single closed fragment (**A**, **B**); and somewhat involved for the faces (**C**) that are composed of both object and cell boundary.

The computation of an object area in a boundary cell consists of two phases. The first one is a simple traverse of all fragments. Closed fragments present no special problem: their area contribution is accumulated as they are encountered. For open fragments (i.e., those that start and end on the cell boundary) both boundary crossing points are stored in a table, which lists the point type (start or end), fragment identification (in form of a pointer), cell neighbor index of the crossing point, and the distance from the closest "upstream" cell vertex. If the cell across the **p**-**q** edge is the first (index **0**) neighbor in the (circular) list of cell neighbor cells, then for fragment **c** two entries are stored in the table: one for entry point **1**, with neighbor index **3** and **s**-**1** distance; and another for exit point **2**, with neighbor index **4** and **t**-**2** distance.

Similar entries would be made for boundary points **3** and **4**, when fragment **d** is processed. In addition, a list of fragments is doubly-linked with the table elements. This ensures that at the end of the traverse of an open fragment, its end-point can be retrieved from the table.

The boundary crossing point table is then sorted, with the neighbor index as the primary and upstream vertex distance as the secondary ordering element. The table in the example would thus be reordered as **3**, **4**, **1**, **2**. This table is considered to be circular, just like the ordered list of cell neighbors.

In the second phase we traverse the faces by starting at the first previously unvisited cell entry (fragment start) point retrieved from the sorted table. The table element is marked as "visited", and the fragment is followed - accumulating the area at each fragment vertex - until the fragment end (cell exit) point is encountered. The table is then searched for the next (in the circular sorted table order) fragment start point. When one is found, the cell vertices (if any) between the two points are traversed and their area contribution is accumulated. If the found entry point has not been visited before, we mark it and traverse its fragment, if it was visited before we have completely encircled a face. Another "unvisited" entry point from the table (if any are left) starts the same process for the next face; if none are left, the boundary cell area computation is completed.

Some Numerical Results

The numerical testing and verification of the area computation presented here differs markedly from that used in Kimerling (1984). There, the comparison is made between the area of a rhomboid computed from its vertex coordinates in UTM projection and the area computed on the ellipsoid using geodetic coordinates of the four equivalent vertices. However, the straight line in UTM projection with a length of almost 250 km (for the largest of the verification figures) is in the general case different from the projection of the geodesic - in this case, the maximum displacement between the two is considerable, and varies significantly between the easterly (11.3 meters) and westerly (28.9 meters) edges. If the geodesic edges connecting the vertices

on the ellipsoid are projected back to the UTM plane (as a sufficiently dense array of vertices), and the area is recomputed, it changes by approximately 1 in 10000 - quite a bit above the precision level of the ellipsoidal area computation claimed by both methods.

Numerical and timing tests performed and presented here use a data object derived from the world coastlines coverage of the Digital Chart of the World (see References, DCW, 1992). It provides multiple "layers" of general-purpose geographical data, commensurate in the precision and density with that of a 1:1 million paper map, using angular geographic coordinates in 5 degree "tiles" on the WGS 84 ellipsoid.

Before the DCW data could be used, however, numerous topological inconsistencies - occurring primarily on the DCW tile edges - had to be detected and removed. The resulting data set consists of 1.3 million vertices, in 27.3 thousand boundary rings - the largest of them encompassing all of the landmass of Europe, Asia and Africa. As a Hipparchus canonical 2-dimensional data set, the object size is slightly over 12 MB. (Raw coordinate data - at 8 bytes per point - takes approximately 11 MB of that).

The average coastline boundary segment is slightly under one kilometer, and less than 1.5% of the segments exceed 3.5 kilometers in length. It is thus safe to assume that the boundary segments - computationally represented by segments of great circles on a conformal sphere - are coincident with the projection of the ellipsoid geodesic connecting the two boundary vertices. (For objects with long boundary segments Hipparchus vector algebra offers a fast yet highly precise approximation of the geodesic line computation as a mid-point between two points: one on the direct and the other on the inverse vertical intersection. (Details of these and other vector-algebra based geodetic techniques can be found in an online Hipparchus Tutorial).

By simply inverting the order of boundary vertices in each ring, we can produce two conjugate objects: one representing the continental landmasses and islands, and the other representing the global Ocean. Both objects are represented in a global Voronoi grid of 2432 cells. The area

of both objects has been calculated, with an obvious expectation that their perimeters (a natural by-product of the computation) will be the same, and that the sum of their areas will be equal to the total planetary surface.

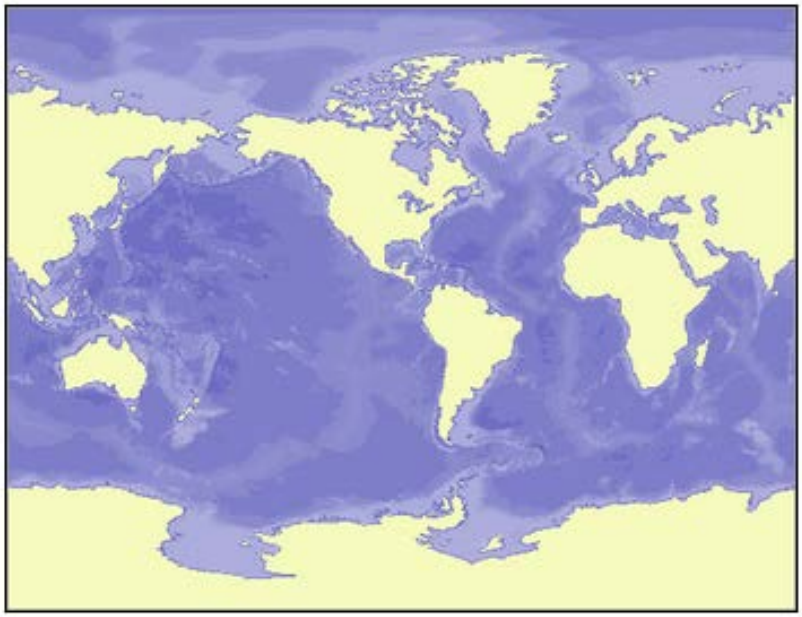


Figure 3: Seven Seas - A Geometry Object

The area computation program initialization consists of the instantiation of the Voronoi grid as a memory-resident object and of the steps necessary to establish the memory-mapping access to the files containing the two objects. The area computation is packaged as a Hipparchus Library function named h7_RsetAreaPerimeter(); its parameters are four pointers to given data: to the Voronoi grid descriptor, some workspace, ellipsoid geometry parameters, and header data of the object for which the area is required; plus two pointers to returned values: the area and the perimeter.

The results presented here have been obtained using the code compiled with the GNU C compiler V.2.95.2, carried out on a 400 MHz Pentium II under Linux kernel 2.2.14. (Performance under NT was only marginally slower).

```
Ellipsoid: WGS 84, a=6378137.0e0, b=6356752.3141e0
```

The 2432 element cell-array area initialization took 0.19 seconds and produced the following values (square meters):

```
Ellipsoid area: 510065621716336.1

Total area of all cells: 510065575723515.5

Difference: 45992820.6

Relative difference (one in): 11090113.0
```

Area computation for both objects took 3.94 seconds (each), and produced the following values (square meters, meters):

```
Landmass area: 150998900960532.0, perimeter: 1249923047.850

Oceans area: 359070890924373.1, perimeter: 1249923047.850

Ellipsoid area: 510065621716336.1

Landmass + Oceans: 510069791884905.1

Difference: 4170168569.0

Relative difference (one in): 122313.0
```

Conclusions

The following conclusions seem to be justified:

- Computation of the ellipsoid cell areas using local mean curvature at the cell center produced the results which are within one in 11 million of the total ellipsoid surface area. For the grid and objects as examined in the example, this is two orders of magnitude better than the results of the object area computation, and thus adequate.

- Computation of two very large planetary complement objects produces results which are within one in 122 thousand of the total ellipsoid surface area. The obtained level of accuracy is a result of a combination of factors; the major one probably the inevitable rounding error in a very large number of (over 1.3 million) of relatively narrow triangles. The obtained accuracy compares favorably with that in many operational systems which compute the areas in the plane - at the outset a significant departure from the true object geometry.
- The very short time needed to carry out the area computations of large objects makes it practical to compute it only as and when this information is required. This facilitates the system design in which the area of a global object of any size and/or shape is not treated as an independent attribute, but rather as only one in a repertoire of measures that can be derived from its canonical geometry definition.

References

1. _____, *Digital Chart of the World* (CD publication), United States Defense Mapping Agency, Fairfax, VA 22031-2137. July 1992.
2. Bomford, G. *Geodesy*, London, Oxford University Press, 1975 ed.
3. Bowyer, A and Woodwark, J. *A programmer's geometry*, Sevenoaks, Butterworths, 1983.
4. Jordan, Eggert and Kneissl, *Handbuch der Vermessungskunde* (Band IV: Mathematische Geodaesie), 10th ed. Metzlersche Verlag., Stuttgart, 1959
5. Kimerling, A. J.*Area Computation from Geodetic Coordinates on the Spheroid*, Surveying and Mapping, Vol. 44, No. 4, 1984, 343-51.

Chapter 7: A Seamless Global Terrain Model in the Hipparchus System

By Hrvoje Lukatela, Geodyssey Limited
Paper presented at the International Conference on Discrete Global Grids,
Santa Barbara, 26 - 28 March 2000

Lukatela, H. (2000). A Seamless Global Terrain Model in the Hipparchus System. In M. Goodchild and A. J. Kimerling (Eds.), Discrete Global Grids. Santa Barbara, CA, USA: National Center for Geographic Information & Analysis.

Abstract

This paper outlines the steps used to construct seamless global triangular networks similar to the planar TIN commonly used to facilitate terrain modeling and volumetrics. It is based on global coordinates and a planetary surface tessellation using spheroidal Voronoi polygons. The techniques used to extend the surface modeling across the Voronoi cell boundaries are presented. The paper also outlines the strategy used for the efficient retrieval of only those parts of the model that are visible in a transient view, as well as the platform- and projection-independent approach to surface rendering.

Introduction

Representation of the physical surface of the Earth in digital systems is a subject of considerable current attention. As the area of the coverage of such systems increases, it becomes necessary to provide methods to model very large, continuous surface conglomerates in a manner which does not violate the surface integrity (i.e., which does not impose hard partitioning as an artifact of the digital model), but, at the same time, provides an efficient spatial index to a small section of the surface of transient interest.

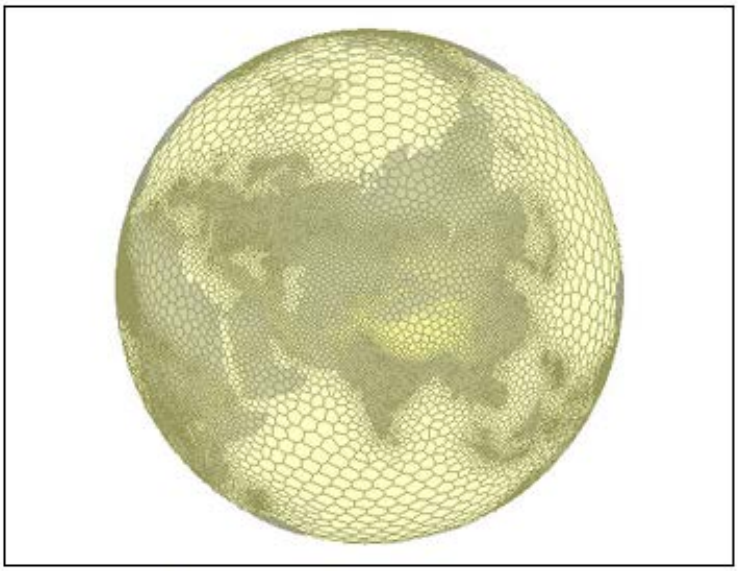


Figure 1: A "global grid" of Voronoi polygons

The Hipparchus Geopositioning Model (outlined in considerable detail in a web-resident publication titled: Hipparchus Geopositioning Model: an Overview) provides a method of construction and manipulation of geometric objects of various levels of complexity (points, lines, areas and surfaces) in a manner which imposes no restrictions on their spatial size or shape. The system is based on two major computational geometry constructs: use of the vector components of the ellipsoid normal instead of the latitude and longitude angles and data-density driven tessellation of the planetary surface using a global grid of spheroidal Voronoi polygons.

The position of a point on the surface of the ellipsoid is best given by a numerical definition of the normal to the surface at that point. The common global coordinates - latitude and longitude - are angles: the first of the two is between the normal and the Equatorial plane and the second between the Equatorial plane projections of the normal and the projection of the prime meridian. Traditional geodetic computations (for instance: given are two known points, determine the length and the azimuth of the shortest line that connects them...) are based on trigonometric functions of those angles and on the expansion of the power series of the eccentricity of the ellipsoid. The angles, however, present two problems when used for computations on digital computers: transcendental functions (sines, cosines) require many

more CPU cycles than the algebraic primitives (addition, multiplication) and their areas of singularity must be compensated with complicated and error-prone code. Thus replacing angles with vector components of the ellipsoid normal was noticed as potentially beneficial as soon as the digital computers were incorporated into the geodetic practice (cf. Bomford (1975), remark on formulae "symmetrical and better for computation...", p. 593 - under 'Cartesian Coordinates in Three Dimensions'). Hipparchus computations are consistently based on the ellipsoid normal given by its vector components instead of the latitude and longitude angles.

A "global grid" is, in the most general sense, a geometrical subdivision of the planetary surface which assists in the organization (partitioning, indexing, etc.) of globally-distributed digital data. "Regularity" of the grid usually translates to simple data structures and straightforward classification algorithms. Since no regular and isometric tessellation of the sphere (beyond five platonic solids) is possible, practical applications have two alternatives: to retain in the design of the "grid" as much "regularity" as possible, to be rewarded with a considerable algorithmic elegance, but at the expense of the ability to perform fast and accurate geodetic computations (see Dutton, (1999) for a well-known example of such approach), or to accept the "irregular" nature of the grid while attempting to make computations based on it as fast and as precise as possible.

The Hipparchus system makes no attempt to produce a "regular" grid. Instead, the grid is designed so that any particular implementation of the grid can match the density of the data that inhabits it and so that the parameters which numerically define the grid "cells" assist (and not hinder) the speed and precision of geodetic computations and spatial algebra productions. (Figure 1 shows a sample of such grid with the density derived from the density of the human population). It is based on the surface subdivision known as "Voronoi polygons": a purposefully selected finite and discrete set of "cell-center" points subdivides the surface so that any surface point is uniquely associated (i.e., it "belongs to") the member of the set that it is closest to. The point classification is accomplished using only distance calculations. Point, line, area and surface object sets are in turn defined in terms of the cells they occupy and the vertex coordinates,

represented not by a the full (global) vector representation of the ellipsoid normals, but by the vector difference from their respective cell center points. An extended introduction to and computational geometry treatment of the Voronoi polygons can be found in O'Rourke (1994).

This paper outlines the canonical representation of the surface object, and explores the techniques which are used to operate upon it. While the context and the examples used in this paper center on the representation of the terrain, such objects can be used to represent any continuous function for which a sufficiently dense set of point-location dependent scalar values are known, and about which we know enough to postulate that each planetary surface point will have one (and only one) measured or interpolated function value.

Triangulated Irregular Network - TIN

The triangulated irregular network (TIN) is an often used surface representation method in planar computational geometry. Description of TIN data structures and the algorithms (including C language source code) can be found in both Ambroziak (1993) and Lischinski (1994). The TIN approximates a continuous surface with a mesh of triangles which more or less coincide with it. The quality of this approximation depends on the combination of the method by which the elevation measurement points have been selected, and the triangulation strategy. The implementation described herein assumes a more-or-less uniform density of significant points and starts with a simple "least-diagonals" fast iterative planar triangulation algorithm. It is open to accept different triangulation strategies - presumably matching more closely the peculiarities of the input data generation process.

The data structure used to represent a planar TIN is simple and shows only minor variations from one implementation to another. It consists conceptually of an array of points and an array of triangles. The two may be doubly linked, but commonly only the triangles are linked to their vertices. Additionally, each triangle is linked to its three neighbors, with a flag value to signal that a triangle edge is at the same time the edge of the TIN (i.e., it is an "outside" edge). Most triangulation algorithms produce a structure in which all outer edges form a planar convex hull.

A point array element consists of planar coordinates and elevation. If the shading or perspective rendering is anticipated, triangle normals might be included in the data; this however would be worthwhile only if the cost of re-computing the triangle orientation far outweighs the cost of additional storage and, usually of even greater significance, the time required to access it.

The main difficulty in the implementation of computational geometry algorithms usually stems from the need to properly predict all (or at least all likely) degeneracy and singularity cases. (See in particular comments under "Robustness" in Lischinski (1994)). While the implementation described in this paper is no exception, it is interesting to note that all such problems were encountered (and hopefully resolved) in the planar ("in-cell") geometry domain.

A TIN in a Global Voronoi Grid

In general, the Voronoi polygon grid used to model the TIN will be used to provide the spatial framework for a number of other object classes. The only requirement that the implementation of TIN data will place on the grid is that the density of the elevation points remains approximately an order of magnitude above the density of the Voronoi polygon centers. In this - and a number of other characteristics - a TIN object in a global Voronoi grid parallels the combined characteristics of Hipparchus point and line sets. This TIN will additionally consist simultaneously of two levels of triangle/elevation data: high-volume triangles with source data points as their vertices and low-volume, large triangles with cell-center and end-points of cell edges as their vertices.

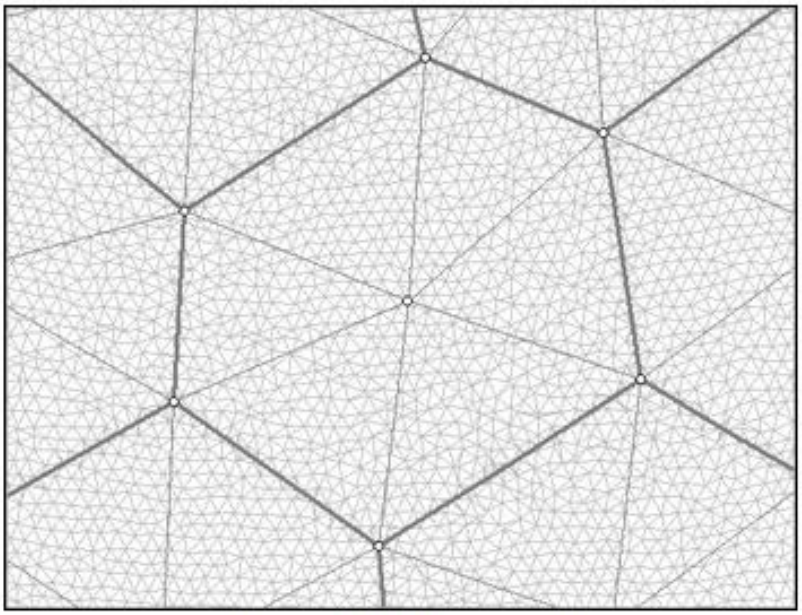


Figure 2: TIN in a Voronoi grid

The source data used in a construction of the Hipparchus TIN object is a Hipparchus point set with elevations. The construction algorithm proceeds cell by cell, transferring the point coordinate and elevation data from the point set into a two-level TIN structure consisting of both individual triangles and the values that describe the elevation of cell-sector triangular vertices. In each cell, the process starts with a selection of all those points that are inside the cell and those points in the neighbor cells that fall within some distance of the shared edge. A triangular grid produced in this step is then intersected with the cell boundary. All triangle edge/cell edge intersection points are assigned an interpolated elevation and marked as "points on the hull". (This can be done, since the Voronoi polygon is at the same time a convex hull of all points in its interior). These points are kept in the final surface representation. Description of planar convex hull and algorithms used to construct it can be found in Sedgwick (1983).

A second triangulation (that includes only cell interior and cell-edge points) follows, this time as a faster, "forced hull" process. The triangles of this second triangulation are stored - together with the point coordinate and elevation data - in a final cell-oriented series of structures representing the surface. Similar to other Hipparchus objects, it is a hierarchical set of tables, at

the center of which is a table of "cell headers". It describes that part of the surface which is "over" the cell, and its most important elements are:

- Cell identifier
- Coordinate array pointer and count
- Count of the cell interior points in the above
- Triangle array pointer and count
- Triangle neighbors array pointer and count
- Minimum, maximum and mean cell elevations

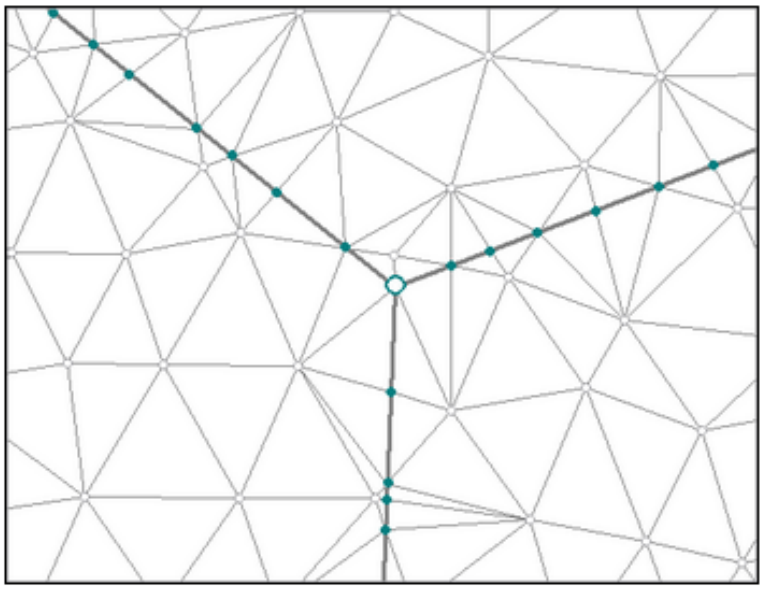


Figure 3: Different types of elevation points

The final TIN point array contains three types of points (clearly distinguished in Figure 3), in order-of-magnitude decreasing numbers:

- Elevation points transferred from the source point set.
- Cell/triangle edge intersections with linearly interpolated elevations.
- Cell vertex points with spatially interpolated elevations.

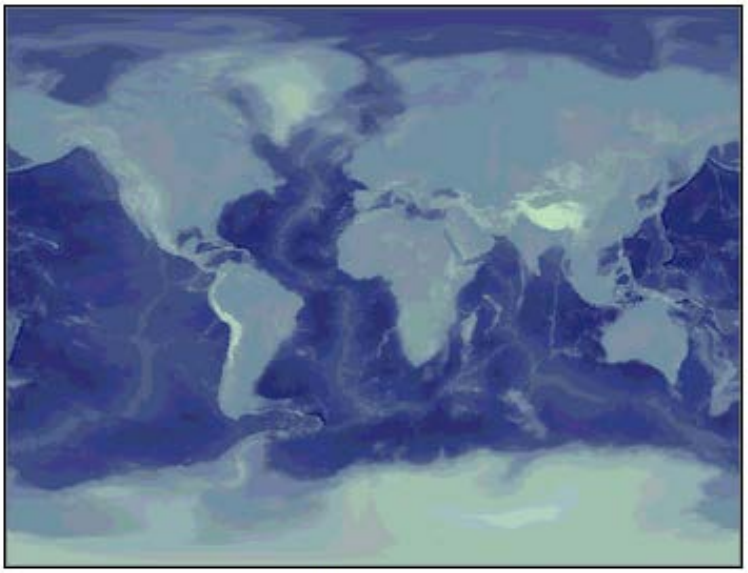


Figure 4: Hypsographic rendering of a global TIN

The test data set (see Figure 4 for its hypsographic representation) used to produce the illustrations of this paper, has been derived from a 5-minute gridded set of elevation points, and used extensively in the development and code testing. A high density Voronoi grid was used as a sampling framework, to avoid an artificially high (and meaningless) data density in high latitudes, and to reduce the point count to a level commensurate with the overall quality of the elevation data. The result of this process was a Hipparchus point set with about 200K points covering the planet in a generally uniform pattern. In addition to its use in the testing, it is anticipated that it will also be used to complete the planetary surface coverage for local (regional, continental, etc.) elevation data sets available in higher density and precision. This strategy is not unlike the frequent use of a Hipparchus Voronoi index center-point set called "isotype", which provides a similar function for regionally-biased grids.

The size of the disk files is approximately 2.5 MB for the source point set, and approximately 8.4 MB as a TIN. In the current implementation, all in-cell triangle indices are 32-bit integers; this removes any practical restriction on the number of triangles in a single cell. Elevations are recorded as 32-bit floating point numbers.

TIN Rendering

TIN rendering is implemented in a series of functions that belong to the general section of the Hipparchus Library dealing with "geographics". A TIN can be rendered in a "reduced dimensionality" mode, to generate only the points and lines (triangle edges) without the full surface representation. This mode was used to generate Figure 3. The Figure and the other illustrations in this paper were created using a scripted geographical workbench program called GALILEO, available for free download from Geodyssey's web page at http://www.geodyssey.com/.

In cartographic applications a TIN is most often rendered as one of the two graphical artifacts: a hypsographic scale color fill or a shaded surface. The first assigns to each pixel a color dependent on the point elevation, the second assigns a color dependent on the spatial orientation (slope and azimuth) of the surface. Figures 5 and 6 illustrate the difference: both show the map background-fill "layer" generated from a global TIN for an area of the European Alps. Hydrographic network and political boundaries are also shown; both are expected to be - to some extent at least - in an easily verifiable relationship to the surface elevations.

Of the two, hypsographic scale color fill is more demanding of the graphical programming, as it requires painting of pixels that belong to a single triangle with a range of colors. (As opposed to shading, where all the pixels that represent one triangle are in general of the same color). We will therefore in the following concentrate on the hypsographic scale TIN rendering.

The central computation geometry process used in this type of rendering is the ubiquitous triangle "gradient fill": given the x and y (image) coordinates of three vertices and their associated "z" value, fill all pixels "inside" the triangle with a color value commensurate to the interpolated value of "z". This interpolated value is a value that a respective point of the plane defined by three points (vertices) would have.

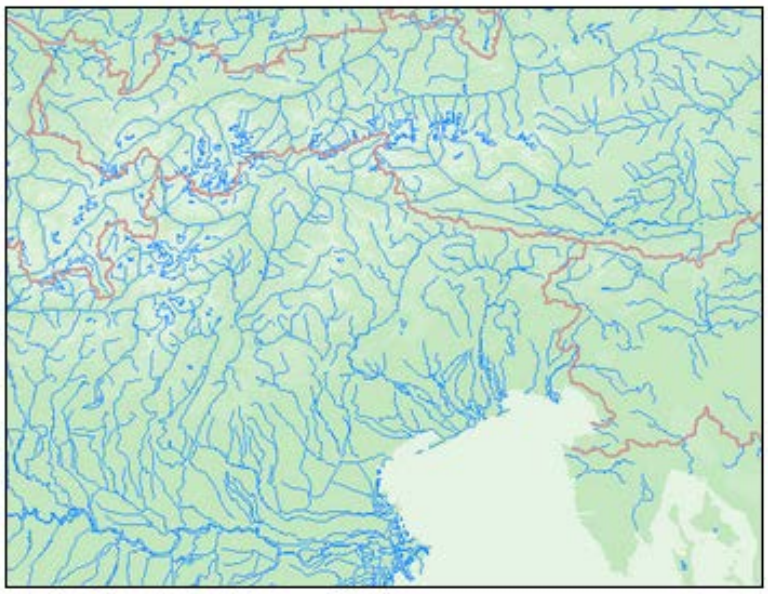


Figure: 5 Hypsographyc rendering

Figure: 6 Shaded rendering

Because of the multiplicity of environments in which this type of rendering is performed, there are different levels at which the division of labor between the application code (including any libraries used), the graphical platform and - possibly - the display hardware can take place. For the Hipparchus Library, we assumed there are at least three such levels, and so provided the means to allocate control to a lower level component (i.e., either the graphical platform or the hardware) at any of those three levels. Their application program interface can be described by the dimension, as follows:

Point

A single pixel is set to a color of the invoker's choice.

Line

Pixels of a scan-line segment with a given y and two end-point x coordinates are set to a linear gradient of colors linearly interpolated between given end-point color values.

Area

All pixels inside a triangle are filled with colors interpolated between the given color values of the vertices.

The first level is available in even the most basic graphical application development environments; the last one is implemented in most "2-D accelerated" graphical device drivers and is supported in graphical platforms targeting such devices. Thus an application rendering a TIN would keep invoking Hipparchus Library functions to the point-level under the Win32 API, and pass a whole triangle to an Open GL API.

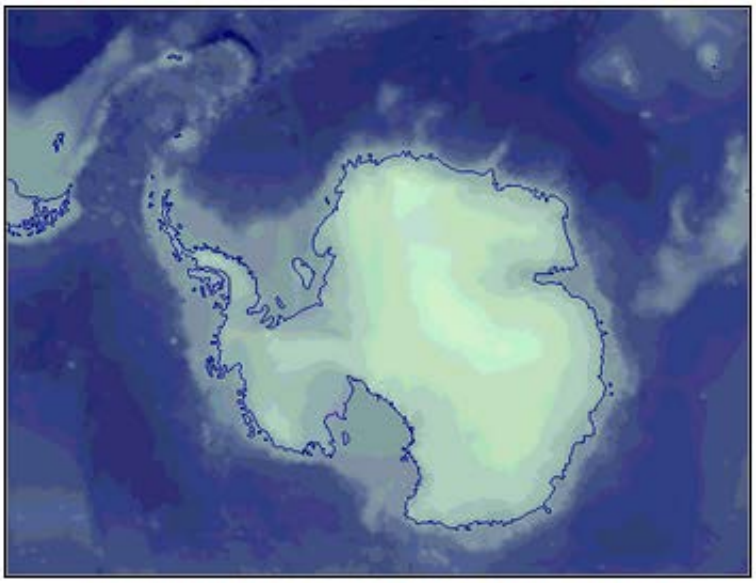


Figure: 7 Sample surface, rendered in (relatively) large scale

A high-latitude section of the global TIN depicted in Figure 7 illustrates another unique advantage of the TIN representation in a Voronoi grid. As mentioned before, all vertices on the cell edge will have an interpolated elevation assigned to them. Likewise, a mean elevation value for the whole cell is a natural by-product of the TIN construction process. This data (cell mean and edges values) can be used as a "generalized" representation of the same TIN: it consists of a set of cell-segment level triangles, probably an order of magnitude fewer per given area than the original TIN. (cf. outlined points and large triangles formed by them in Figure 2). When the scale of rendering becomes sufficiently small, the application can choose to render only the large triangles, thus decreasing the rendering time considerably. Applications will often store Hipparchus Binary Objects (on disk or in a database) so that the low-volume cell-related values are separated from the high-volume point coordinates. If this is the case, rendering based only on the cell-level values (such as depicted on the small-scale map in Figure 8) will require at least an order of magnitude fewer disk transfers than the rendering of Figure 7 - despite the fact that both are generated from the same surface object.

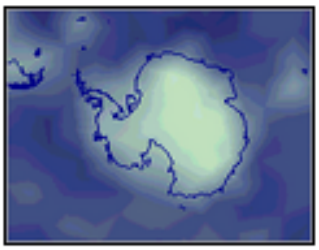


Figure 8: Cell-level generalization

Conclusion

A TIN as a Hipparchus Binary Object shares many architectural similarities with point sets, line sets and regions constructed and recorded in the context of global Voronoi grids. Its construction is based on a planar triangulation within the cell boundaries. Triangle information storage follows the common planar TIN model, but an additional set of elevations is stored to represent the cell edge elevation profile. Simple, straightforward and conjugate linear

interpolation of elevations on the edge guarantees that no artifact will be introduced at the cell edge. The cell-based data structure representing the whole TIN provides a simple and fast determination of the elevation at an arbitrary selected surface point, and triangle rendering fits well with the API of common graphical platforms. A novel approach to the problem of generalized rendering is another benefit of the Voronoi grid.

References

1. Ambroziak, R.A., Cook C.A., Woodwell, G.R. and Wicks R.E.: *Data, Software, and Applications for Education and Research in Geology* (CD publication), United States Geological Survey Open File Report 93-231, 1993.
2. Bomford, G. *Geodesy*, London, Oxford University Press, 1975 ed.
3. Dutton, G. H.:*A Hierarchical Coordinate System for Geoprocessing and Cartography*, Springer Lecture Notes in Earth Sciences 1999.
4. Lischinski, D.:*Incremental Delauny Triangulation*, article published in The Graphics Gems Series, Volume IV, pp. 47-59, AP Professional 1994.
5. O'Rourke, R.:*Computational Geometry in C*, Cambridge University Press 1994.
6. Sedgewick, R.:*Algorithms*, Addison-Wesley 1983.

Chapter 8:

Criteria and Measures for the Comparison of Global Geocoding Systems

By Keith C. Clarke
University of California, Santa Barbara
Santa Barbara, CA

ABSTRACT

There is no shortage of systems for global georeferencing. Each system, however, differs to a varying extent from a hypothetical ideal system and from other systems. Factors of variation are the systems: authority, succinctness, definitiveness, degree of exhaustiveness in coverage, scaling properties, degree of hierarchical structure, uniqueness, intuitive understandability, tractability, and accuracy. These properties are defined, using examples from some existing grid systems, and are developed as criteria against which the comparison of systems is possible. For each criterion, a metric or metrics are suggested that can be taken individually or collectively for comparing global geocoding systems quantitatively. Methods for such a comparison are discussed and presented. It is suggested that user oriented criteria should be weighted more heavily in global grids for the information and g-commerce age.

INTRODUCTION

Within the last few decades, the number of global georeferencing systems available for applications such as navigation, cartography, position finding and surveying has multiplied significantly. A standard cartographic textbook (Robinson et al., 1995) lists the following systems in use in the U.S.: telephone area codes, postal zip codes, street addresses, letter-number grids, geographic coordinates, cartesian coordinates, GEOREF (or the World Geographic Reference System), Universal Transverse Mercator (both the Military Grid and the USGS civilian version), Universal Polar Stereographic, State Plane Coordinates, and the U.S. Public Lands Survey System. Quite clearly, there is no shortage of ways in which to digitally or

otherwise encode position on the earth's surface. Only as recently as 1884, however, was the long used geographic coordinate system finally institutionalized, probably forming the first truly global grid.

In October 1884, at the invitation of the President of the United States, 41 delegates from 25 nations met in Washington, DC, for the International Meridian Conference. Conference resolutions 1 to 3 dealt with standardizing geographic coordinates. It was resolved that "It was desirable to adopt a single world meridian to replace the numerous one's already in existence. 2.The Meridian passing through the principal Transit Instrument at the Observatory at Greenwich was to be the 'initial meridian' and 3.That all longitude would be calculated both east and west from this meridian up to 180 degrees." Resolution 2, fixing the Meridian at Greenwich was passed 22-1 (San Domingo voted against), France & Brazil abstained (Howse, 1997, p. 141). Thus the Meridian conference established two criteria for acceptance of geographic coordinates as a global geocoding system. These were its universality (by determining its scope of application and origin) and authority (by the acts of resolving and voting at an international conference).

This paper examines briefly the criteria that a global geocoding system or geographical reference frame strives for. These criteria are recounted here not out of originality, but with the intent of providing a framework against which individual geocoding systems can be compared. Furthermore, since qualitative comparisons serve primarily to stimulate debate rather than good science, a set of metrics is proposed so that each existing and new system alike can be measured against the remainder and against definitive absolute metrics. To illustrate the approach, these metrics are computed for a selected small set of existing global grid systems and the results presented for discussion. It is hoped that presenting the metrics and their means of computation will stimulate analytical studies of global grid systems, to refine their applicability and characteristics, and to aid the map user in their selection for a particular purpose.

DIMENSIONS FOR COMPARISON OF GLOBAL GEOCODING METHODS

To this author's knowledge, there have been at least two prior attempts to derive criteria for comparing global grids. Goodchild (1994) listed 14 criteria, set forth in Table 1. Kimerling et al. (1999) added to, refined and reordered Goodchild's criteria to reflect their work on nested hierarchical tessellation. Kimerling et al. (1999) saw an ideal global grid as being able to summarize irregular global measurements, calculate gradients, compare time series, compare regions, compare data collected at different resolutions, improve numerical modeling and to document the precision and location of spatial data on the globe. Criteria used for evaluations of the grid systems were chosen analytically as further derivatives of the Goodchild criteria. Those of critical importance were metrics for spheroidal area, compactness and center point spacing. Some specific metrics were proposed that included the Zone Standardized Compactness. In addition, Kimerling noted that some criteria could be automatically tested, for example by point-in-polygon testing, coordinate range checking, and verifying that recursion is possible.

Table 1: Comparison of Criteria for the Assessment of Global Grids

Criteria in Goodchild (1994)	***Criteria in Kimerling et al. (1999) (Goodchild's Numbers given in parentheses)***
1. Each area contains one point	Areal cells constitute a complete tiling of the globe, exhaustively covering the globe without overlapping. (3,7)
2. Areas are equal in size	Areal cells have equal areas. This minimizes the confounding effects of area variation in analysis, and provides equal probabilities for sampling designs. (2)
3. Areas exhaustively cover the domain	Areal cells have the same topology (same number of edges and vertices). (9, 14)
4. Areas are equal in shape	Areal cells have the same shape. ideally a regular spherical polygon with edges that are great circles. (4)
5. Points form a hierarchy preserving some (undefined) property for m < n points	Areal cells are compact. (10)
6. Areas form a hierarchy preserving some (undefined) property for m < n areas	Edges of cells are straight in a projection. (8)
7. The domain is the globe (sphere, spheroid)	The midpoint of an arc connecting two adjacent cells coincides with the midpoint of the edge between the two cells.
8. Edges of areas are straight on some projection	The points and areal cells of the various resolution grids which constitute the grid system form a hierarchy which displays a high degree of regularity. (5,6)
9. Areas have the same number of edges	A single areal cell contains only one grid reference point.(1)
10. Areas are compact	Grid reference points are maximally central within areal cells. (11)
11. Points are maximally central within areas	Grid reference points are equidistant from their neighbors. (12)
12. Points are equidistant	Grid reference points and areal cells display regularities and other properties which allow them to be addressed in an efficient manner.
13. Edges are areas of equal length	The grid system has a simple relationship to latitude and longitude.
14. Addresses of points and areas are regular and reflect other properties	The grid system contains grids of any arbitrary defined spatial resolution. (5,6)

This study hopes to build upon this pioneering research. In Kimerling et al.'s rebuilding of the criteria, two factors emerged. First, that metrics are critical to implementing and using the criteria effectively. Second, that while commonalties have emerged between the two versions of the Goodchild criteria, nevertheless the Kimerling et al.'s work was oriented primarily toward hierarchical recursive global grids. An attempt is made here, therefore, to be both more specific in terms of metrics, and more general in terms of criteria. Furthermore, criteria that relate to geometry rather than topology (such as Kimerling's seventh and Goodchild's eighth) are specific to projection properties rather than grid characteristics. In many grids, a projection is assumed, and recursion takes place on the plane. The grid's relation to projection distortion is then a given. This is because Goodchild's properties 2, 4, 8, 9, 10, 12, and 13 are consequences of the projection decision and the assumption of the earth model, not necessarily the grid systematics. This is the case even with geographic coordinates.

The approach taken here is more generic, and includes more of a grid system user orientation than an algorithmic, topological, or computational geometric perspective. It is hoped that a user based grid comparison will be of use in the grid selection process, and assist users in learning grid systems. First, the dimensions of the criteria are considered, and some metrics identified. These are collected into a single framework, and some examples given for specific grids.

1. Universal

Global grids are designed to be universal. That is, in the ideal, they apply not only to all three dimensional bounded objects such as the geoid and the planets, but also to the whole earth. Clearly a first assumption affecting universality is the choice of earth model. Historically, geodesy divides chronologically into periods based on the sphere, the oblate ellipsoid and the geoid as earth models. Since the geoid is usually expressed as deviations from the best fit ellipsoid, and since the earth fits the spherical model reasonably well for highly generalized mapping applications, the universality dimension clearly reflects accuracy. At the overview level, however, universality may be thought of as the degree to which the global grid system allows location georeferencing for the whole earth or equivalent geographic object (such as the Moon, Mars, or a baseball).

Related to the universality dimension is the ability to recover the system from empirically derivable features. Examples are the poles, the equator and universal time, all of which are absolute and measurable given the right algorithm. Actual origins, however, may or may not be tied to tangible features. Perhaps the most universal of systems is geographic coordinates, which applies to all three earth models and to the whole planet. Nevertheless, its origin point (Figure 1) is recoverable only with extensive use of sophisticated navigation equipment and/or astronomical observation. A first metric of universality, therefore, might be the degree or ease of recoverability of the primary reference monuments and the linear dimensions for the system.

Figure 1. Origin Point of the Geographic Coordinate System (0 degrees North, 0 degrees East)

On the purely practical level, universality also relates to the adoption of standards. International Standards Organization (ISO) standards represent the peak of a hierarchy that moves through national standards such as ANSI, all the way to "industry" standards and de facto standards such as PostScript. Some measure of universality, therefore, should reflect the incorporation of broadly acceptable standards. This may include the use of the decimal system, use of metric units, use of Arabic numbers (0,1,2,3... etc.), and the specification of a reference ellipsoid by an international body. For example, WGS84 is broadly accepted and used as a reference ellipsoid, though its exact specification remains classified. A higher standard is the International Earth Rotation Service's International Terrestrial Reference System (ITRS). A second measure of universality therefore, might be the number of geocoding system parameters that are tied to standards, weighted for the level of the standard.

In addition, universality applies to the extent of the system's actual coverage. In UTM, for example, coverage extends from 84 degrees North to 80 degrees South, leaving the UPS for polar coverage. While UTM covers just about all of the earth's inhabited area, nevertheless the real extent is incomplete, depending on the UPS to fill the gaps. A metric corresponding to the extent of coverage might be the actual land area of the terrain "covered" by the grid, as a ratio to the total surface area of the earth model or ellipsoid. This metric is reconsidered under the dimension of exhaustiveness.

2. Authoritative

Almost anyone is capable of devising a global georeferencing system. Not all, however, are of equal credibility. Systems establish authority in two ways, by recognition and by acceptance. Recognition implies a hierarchy of acknowledgment. At one extreme, a system "exists" if it is in any way documented. So for example, an independent scholar could publish a minimal set of definitional parameters as a web page, and the system would exist but with no recognition. An example of such a system would be the geographic coordinate referencing using Washington, D.C. as the prime meridian, marked on many older American maps. Initial recognition of a system is primarily academic or highly specialized, but publication in the peer reviewed scientific literature of cartography, as in the case of Dutton's Quaternary Triangular Mesh (Dutton, 1999) is a critical mark of authority. Passing from academic research into standard teaching practice also enhances credibility, so that inclusion of a system in a cartography textbook, or its teaching at a major University in any relevant curriculum adds a further degree of credibility (and acceptance). A metric of such inclusion might be the number of references in a bibliography, the number of citations to the system, or the number of entries in catalogs or textbook indices.

Formalization into standards, such as FIPS 173, at the national level moves the system to the next level of credibility. Beyond national standards lie those of international collaboratives (e.g. scientific or professional societies, NATO), then the official recognition of International Standards bodies and the United Nations. The so-called Peters projection, for example, was less than a historical footnote until Peters convinced the World Council of Churches and the United Nations to endorse the map (Monmonier, 1995).

At the highest level of recognition lies acceptance by the International Standards Organization (ISO), and by consortia of professional societies at the international level and acceptance for extended periods of time. For example, the International Earth Rotation Service (IERS) was created in 1988 by the International Union of Geodesy and Geophysics (IUGG) and the

International Astronomical Union. It replaced the Earth rotation section of the Bureau International de l'Heure, and the International Polar Motion Service. IERS is a member of the Federation of Astronomical and Geophysical Data Analysis Services and has been established since 1988 to provide to the worldwide scientific and technical community reference values for Earth orientation parameters and reference realizations of internationally accepted celestial and terrestrial reference systems. The IERS is charged to define, use and promote the International Terrestrial Reference System (ITRS) as defined by the IUGG resolution No 2 adopted in Vienna in 1991.

Another type of authority is that of broad popular acceptance and use. Map users are highly influenced by choices make by the map production agencies. The United States Geological Survey, for example, normally marks geographic coordinates, UTM, and State Plane coordinates on the collars of its topographic quadrangle maps at 1:24,000. City Street Guides commonly use one-off Alphabet-Number referencing. Neither group based the choice of systems that the map consumer must use by practical necessity on user demand assessment, although the USGS has occasionally changed grids and their depiction at the request of its fellow federal agencies such as the US Forest Service. Acceptance of systems is probably highest with street referencing, involving common use and standardization by the US Post Office, and including extensive international cooperation.

Measures of authority are necessarily subjective, since they are based primarily on trust. A simple measure would be the hierarchical level in the schema from individual to global organization. Another would be the customer acceptance, in terms of market share. A simple alternative metric would be the number of standards that form the definition of a particular grid system, or conversely, the number of standards that use a system for defining geographical referencing.

3. Succinct

The ideal grid reference results in coordinates that are terse. Typically, a grid reference handles the three dimensions separately, with each depending upon one or more items of metadata that define a zone or region. The degree to which the elements of the coordinate are embedded are important. Systems exist which contain the entire global reference in the string, use separate strings for eastings and northings (and elevations), and interleave the digits of the reference in alternating pairs.

Communication theory provides a measure of the quantity of information that flows during a transmission from sender to receiver (Shannon, 1948). Transmission flows involving coordinates are common, for example when GPS data are collected, when search and rescue operations are conducted, or when GIS coordinate based data are moved between systems or used over a network. The general formula for reducing the uncertainty of communication to zero ("computing the amount of information generated by the reduction of n possibilities (all equally likely) to 1." Dreske, 1999) or the entropy, is given by the base 2 log of n. This value, in bits, is the necessary length of a binary number that fully defines an atomic unit in the system undergoing transmission. In the case of coordinates, this is a point. We will consider only the case of two coordinates, in spite of the fact that three coordinates strictly are necessary to define location (the reference ellipsoid is assumed instead of the geoid).

A paradox of grid coordinate systems is that points located close to each other in geographic space have similar coordinates. This is a corollary of Tobler's famed "first law of geography," that everything is related to everything else, but near things are more related than distant things (Tobler, 1970). If indeed the importance of discrimination of locations is equally important as points become closer together, say for local navigation, then the redundancy within the coordinates becomes a maximum, when it would be best be minimum.

Tukey (1977) advanced the stem-and-leaf plot as a simple tool for finding the level of redundancy in transmissions. Shannon (1948) had previously devised a mathematical theory of

information in communication that related the amount of information transferred to the reduction of the uncertainty at the receiver end of the transmission. Such an approach can be used with coordinates in a global grid system. For example, as a map user parses along a vector, information is sequentially extracted from strings of coordinate pairs that carry geographic information content, usually coded from left to right, but sometimes interwoven and alternating digits. As each successive digit (or bit) is traversed, more spatial information flows, and serves the function of distinguishing the current point from the previous point. Similarly, we can talk about the set of points constituting a geographic feature, whether it be a point, line, area or compound feature.

For a set of geographic coordinates, a ratio can be defined that is the measured range, standard deviation, and maximum possible range of each coordinate. For example, for the set of coordinates {632794.69 4538257.50 632948.69 4538520.00 632554.25 4538098.50 632794.69 4538257.50 632231.31 4537331.00 632554.25 4538098.50} which constitute UTM coordinates in Zone 18, Northern hemisphere, forming a small part of the outline of Long Island, we can compute the following values:

Table 2. Statistical Description of Six Coordinate pairs in UTM

Minimum (x, y)	Maximum	Range (m)	Standard Deviation (m)	Std. Dev as Proportion of Range	Zone 18, N
632231.31	632948.69	717.38	232.609114	0.32425	UTM Easting
4537331.00	4538520.00	1189.00	369.041521	0.31038	UTM Northing

These estimates of total range and variance, however, mask the variance structure as it relates to the coordinates digits themselves. Accordingly, consider that for every significant digit of the coordinate there is both an actual and an expected proportion of the coordinate count within the set. For the ten digits of decimal numbers, if over the set the actual proportional occurrence of each digit, "0", "1", "2" and so forth was the same as the expected (0.1) then the sum of the deviations would be zero. If all coordinate digits were identical in occurrence across points, then nine digits would have no occurrence (0.0 - 0.1 x 9 = -0.9) and one digit would have the whole point set (1.0 - 0.1 = 0.9), whose magnitudes sum to 1.8. This is the case for the first

three digits of both the easting and the northing in the set. Thus digit variances would vary from zero (for no difference between expected and actual digit occurrence) and 1.8, when all digits are identical. The equivalent values are 1.875 for hexadecimal, 1.75 for octal and 1 for binary, given by *2 (B-1)/B* where *B* is the number base. For any digit *n* at any one significant digit location out of *N* possible digit values or states (10 for decimal), *I* is defined, where:

$$I_n = \sum_{1}^{N} \left| \frac{D}{\Sigma D_n} - \frac{1}{N} \right|$$

Similarly, the set of these values across all significant digits defines a function that starts at complete redundancy, drops as the information content increases, then returns to near redundancy in this case beyond the decimal point. This might be termed the *Coordinate Digit Density function*. The "area" or total divergence of this function from complete redundancy defines a value which is an entropy or information quantity value for the coordinate set which may be independent of the coordinate system and therefore of value in comparison. This value, termed *S*, for *N* significant digits and assuming number base *B* is given by:

$$S = \sum_{1}^{N} \left(I_n - \frac{2(B-1)}{B} \right)$$

Computing *S* for the six point set above yields 2.167 for the northings and 2.407 for the eastings. Both eastings and northings have their highest maximum entropy at the third decimal place, and have five redundant digits. The Coordinate Digit Density functions are shown in figure 2.

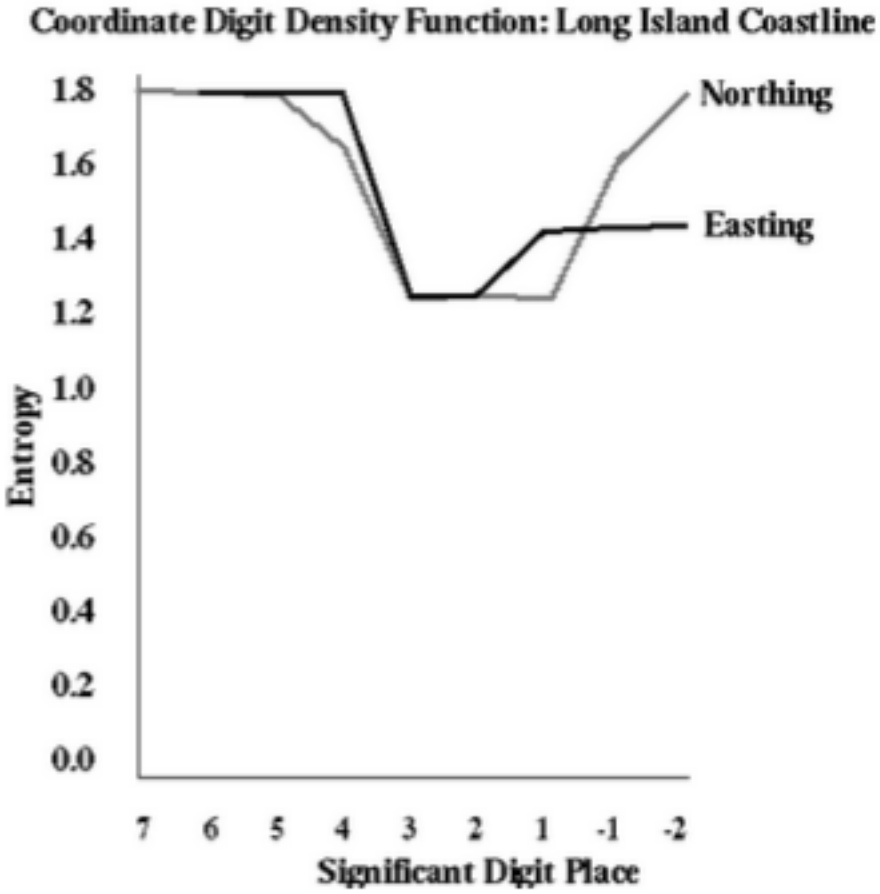


Figure 2. Coordinate Digit Density Function

The easting and northings show some differences, which for a larger point sample may be worth investigating. For example, neither reaches a value below 1.0, which might be expected at the peak information content digit near the decimal place. The first three and last digits of the northing and the first three digits of the easting are redundant. Any compression system, such as leaving off the zone number and a 100 km intersection reference and truncation of redundant decimal places would compress the data to those digits below the 1.8 line. Significantly, the abbreviation of the northing to the nearest 0.5 meter and the repetitive rounding of the northing imply that the data have been converted from other units or degrees (they were).

The measure *S*, and the Coordinate Digit Density function are proposed as useful means for the comparison of global grid systems. Since the density function depends only on frequencies, letters and other systems, such as octal, binary and hexadecimal are equally suited to the metric. Comparison is possible between and among the same exact position in different systems, set of positions in different systems, or coordinate extremes. Simple statistical description implied that the northing had a much higher variation in range and standard deviation, but a lower deviation as a proportion of the range. The digit analysis shows that the eastings have a higher information content, and that it is concentrated between the fourth digit and second decimal place, i.e. in the 1km to 1cm range.

A succinct coordinate system, therefore, is one with the most spiked coordinate digit density function, the lowest entropy associated with a single coordinate digit, and the highest S. Nevertheless, this ignores the fact that many coordinate systems are hierarchical. While the measure applies equally well to almost all systems, it is possible to score the levels of the hierarchy independently, to measure the bits required per level (using Shannon's formulae), and to estimate the total information content of a set of coordinates in different systems.

While succinctness depends heavily on assessing the total amount of information in coordinates, there are at least two other measures that may be critical for comparing systems. The first of these is retrieval time, that is the amount of time it takes to move an encrypted coordinate into a system where it can be readily interpreted. This can be considered in several ways. At the simplest, it is a simple number of algorithm steps, computations or look-ups necessary to write a coordinate into ASCII digits. So, for example, a transformation of coordinates may involve a decompression, binary to decimal conversion, bounds checks and an affine transformation. This could be quantified as a number of steps or as real CPU seconds. Secondly, use time is the opposite transformation as applied to the user of the grid system. How long would it take a novice or skilled map user to place the point onto a map, or encode a given location? This could be assessed subjectively, or measured in real seconds by testing map users.

4. Definitive

Definitiveness is the ability of a coordinate or grid system to unambiguously determine a georeference. There are at least three concerns for definitiveness. First, a single point on the earth's surface must be assigned one and only one reference. Clearly this is often not the case in many global grid systems. The geographic coordinate system assigns identical latitudes and both marginal and extreme range longitudes to a single point at the poles. UTM allows a half degree overlap between zones, a fact that becomes vital when regions of interest fall over zone boundaries. A quantitative metric of overlap definitiveness for a whole system could be the total earth surface area for which overlap is permitted, perhaps weighted for the total number of overlaps (there may be more than two).

Lack of overlap reduces redundancy, but may be an integral part of the grid system. Equally as important to the amount of overlap is the lack of confusability between higher orders of coordinates. For example, within a UTM zone (and even between systems, such as USPLS in meters) there is little to distinguish between zones. The Military Grid system recognizes this by assigning letter references and redundant discriminators to the UTM zone number, which usually has only one digit offset. Thus an effective grid system never creates coordinates for nearby points that have lower order northing and easting coordinate specifications that are similar. This argues for integrated hierarchies in coordinates, so that the confusion is explicitly eliminated. It is also supportive of those georeference systems that interleave eastings and northings.

Similarly, a global grid system should be able to support objects of different dimensions. Here the largest difference is the treatment of the coordinates themselves. In the Military Grid and the British National Grid, the reference unit is of varying area depending on the interwoven number of digits employed. While this works well for many applications, it is usually not sufficient for mixed point, line and area objects.

5. Exhaustive

Opposite to definitiveness is exhaustiveness. A grid system should cover each and every location on earth at any level of scaling, spatial resolution, or measurement precision. Complete exhaustiveness is less common than might be thought. UTM for example, covers only 80 degrees South to 84 degrees North. Just as we measured universality as a proportion of the earth's surface area, we can similarly quantify exhaustiveness as the proportion of the earth's surface covered by the system.

Exhaustiveness may or may not scale. For example, UTM is not exhaustive on a global level, even within a zone, but has redundant coverage at the edges of zones. Thus exhaustiveness should be assessed both at the global level and the local level in a hierarchical system, perhaps at every level. Similarly, grid exhaustiveness may be a function of the resolution of the atomic grid unit. Obviously as resolution becomes coarser, assigning grids to features involves overlap, redundancy and drop out. Some measures of this exhaustiveness have been quantified by Mulcahy (1999).

Finally, a grid system should be able to store sufficient precision to ensure exhaustiveness. At least, this means that atomic features (for example, bench marks, survey points, pixels on high resolution images, utility features such as power poles and manholes) must have a unique location. A better condition might be that the precision associated with the atomic unit for the grid is close to the accuracy of the measurement instruments. The latter implies an "effective resolution" that might be concisely delimited using the sampling theorem (Tobler, 2000). As a rule of thumb, the grid "spacing" or level at which precision is capable of feature discrimination should be less than half the average size of the smallest feature that the system is designed to locate. There are standard measures of precision and accuracy (Goodchild and Gopal, 1989). Relating these to the grid's effective atomic unit via the sampling theorem might be best done with a simple ratio.

6. Hierarchical

The merits of recursion and the repetition of rules and structures inherent in hierarchies are too great to be ignored with global grid systems, and almost all systems use some degree of hierarchical tessellation or tiling. Nevertheless, tiling of any sort creates a tension between core and edge. Typically tiles are reprojected using unique central meridians, points of tangency or secancy, so that the pattern of error is centered on the tile and is maximum at the edge. Examples are shown in Kimerling et al. (1999).

Joints are where tiles meet on the ground. If joints overlap, tiles or zones interleave to form zones of a lack of definitiveness and redundancy. The geometry of the overlap and joints is important for accuracy, scale, direction, area and shape on the grid. These properties have long been quantified and even cartographically symbolized on maps (Mulcahy and Clarke, 2001). A metric of jointing should simply reflect the amount and distribution of joins in the system. This is given by (1) the number of recursive tilings used to reference the atomic unit in the grid and (2) the total number and length of tile edges in the entire system. This refers to all appropriate unique hierarchical levels, and may be computed as a function of level. The Quaternary Triangular Mesh (Dutton, 1999) for example, does not vary with recursion beyond the level one partition into triangles from the globe, nor does the quad tree approach of Tobler and Chen (1986). UTM has only 60 interior and 120 exterior edges at the highest (zone) division. Thus UTM could be said to score 180 on the zone edge scale.

7. Unique

The degree of uniqueness has already been covered under definitiveness and exhaustiveness, and is part of both Goodchild and Kimerling's criteria lists. The problem of coordinate confusion is a real one, both within and between systems, and perceptual testing could be used to define uniqueness in terms of user errors in coordinate interpretation. Some notorious errors in coordinate specification, from Embassy bombings to friendly fire incidents, could be easily avoided with effective enforcement of local uniqueness of coordinates. To be unique, each

entity or atomic spatial feature has only one geocode, and the geocodes are distinctive from each other. A measure may be the average number of significant digits that discriminate between features deemed to be adjacent or contiguous. Such an assessment would be possible by measuring the Coordinate Digit Density function for point pairs or point sets drawn at random from points at "near" and "far" distances from each other. A correspondence measure between two coordinate pair sets could be simply the proportion of coordinate bits that match exactly divided by the sum of the match plus the mismatch.

8. Intuitive

Anyone who has taught global grids and coordinate systems in undergraduate cartography or geography knows that many people find the grasp of the basics of grid systems an intellectual challenge at best, and a mystery at worst. There is a strong correlation between the ease of teaching a system and the system's effective use in practice, especially in applications such as field survey and navigation. Simplicity, however, is a highly subjective metric. Occam's razor tells us that if there are two acceptable theories explaining a set of facts, the simpler one is better. Such a rule can be applied also to global grids, yet with caution, since the intended function of the grid system is every bit as critical to effectiveness as simplicity of the system's rules and constants.

Measuring intuition is perhaps hardest of all of the measures proposed! One set of ways of quantifying intuition is to count the impediments to use of the system. Possible metrics are (1) the number of separate "facts" necessary to learn or explain the system; (2) The number of "magic numbers", i.e. constants, arbitrary origin points, earth radii etc. necessary to define the system or to locate a single point within it; (3) The average number of words or pages that a software manual, textbook, or help system must devote to explaining the system to a user.

Harder to quantify are metrics that define the time necessary for fluency in the system (say, to achieve making no errors per 1,000 point fixes) by book learning or experience. Among these are the level (elementary school, junior high, high school, college) at which education is

possible, the reading grade level necessary to understand system documentation, the time required for explanation of the grid, and the amount of retraining required for maintenance of the knowledge.

Another important property of global grids is their memorability. This property is not the memorability of the systematics of the system, but the memorability of the georeferences themselves. For example, it is relatively easy to remember that Santa Barbara lies in UTM Zone 11, and that the boundary of Zone 10 is at the 120th meridian just West of Goleta, but only if you live in Santa Barbara! Other aspects of grid references may or may not easily commit themselves to memory, but if they do are useful for all sorts of basic fact retrieval and navigation. Effective systems promote such recall, and exploit it. Nevertheless, measurement of this property seems almost impossible without resorting to qualitative methods.

9. Tractable

Many advantages of global grids are not necessarily part of the system but are consequences of the system's properties. Central to the tractability of a global grid system is the availability of a mechanism for encoding, decoding and plotting of the systems with maps. This may imply web access, computer programs, software utility or built-in functions, and full documentation. A measure of the tractability and ease of use of a system is the programming code volume in bytes, number of logic steps, number of lines of code, or program execution time associated with deriving or plotting coordinates of different types.

Secondly, many applications of coordinates focus on their use for the extraction of successive sample locations directly from random numbers applied to the coordinates and their ranges. To be suitable, a grid system should allow the extraction of samples in random, systematic, hierarchical or other appropriate sampling methods. Resampling coordinates is often an important part of multi-scale cartography, therefore support for multiple representations or multiple display scales is desirable. If this happens as coordinates are resampled, then generalization is possible by simply weeding duplicate coordinates.

10. Accurate

Traditional metrics of accuracy involve tests against independent map sources of higher authority. A measure of a grid's accuracy, in addition to the already suggested ratio of resolution to features size, is the comparison with an independent, or original source. If a common database, such as the Digital Chart of the World, is transformed to another grid system, and then the transformation is inverted, then there should be a one-to-one correspondence on a bit-by-bit level between the original and retransformed maps (Clarke, 1995). Any disagreement, as a proportion of the original, is the omission error. Such error can be quantified in many ways.

Within a system, accuracy is defined by repeatability. A grid system should be able to return a user or navigator to the exact same location, independently of minor details such as rounding error, algorithm implementation, and pixel resolution. A measure might be to locate a set of points one thousand times each, and to quantify the average positional error involved in repetition. Even with computer algorithms differences emerge. With human interpretation and with look-up solutions, errors can be significant.

Finally, error is both a global and a local property of grids. Aggregate accuracy measures mask the extremes and spatial distribution of error. Efforts should be made to portray not just the amount of error, but also its spatial distribution. This is often possible with quite traditional cartographic methods (Clarke and Teague, 1998).

METRICS FOR COMPARISON

Several metrics have been proposed in the discussion of the dimensions of comparability. Any or all of these could be computed and used for comparison between global grid systems. In Table 3, attention has been given to the ranges and units of the metrics. Most values could be computed, of course, in several different ways.

Table 3. Summary of the Metrics for Global Grid Comparison

Dimension	Metric	Value	Units	Geographic Coordinate Example
Universality	Proportion of earth's surface covered by grid	Ratio (0.0-1.0)	None	1.0
Universality	Recoverability of grid system origin monument and standard dimensions	Boolean (2 values)	Yes/No	{0, 1} Units are ISO defined
Universality	Proportion of parameters and constants tied to International or national standard	Ratio (0.0-1.0)	None	1.0 Assuming metadata (e.g. ITRF, WGS84)
Universality	Number of International or National Standards referenced by specification	Greater than or equal to zero	Standards	2 (Ellipsoid, International Meridian Conference)
Authority	Number of bibliographic references to grid system	Greater than or equal to zero	References in Snyder bibliography	351
Authority	Number of references on World Wide Web	Greater than or equal to zero	Number of hits using altavista with "geographic coordinate"on 3/21/00	1,305,095
Authority	Number of catalog entries	Greater than or equal to zero	Search of "magazine and journal articles" in all UC libraries on 3/21/00 for "geographic coordinates" under subject keywords	27
Authority	Entries in textbook index	Greater than or equal to zero	Number of pages referenced in index for Sixth Edition of Robinson et al. "Elements of Cartography", under "geographical coordinates", "latitude", and "longitude"	11

Authority	Degree of conformance to standard	Boolean	Assumed, since ISO references.	1, Yes
Authority	Number of standards that reference the system	Greater than or equal to zero	Standards	Unknown
Succinctness	Number of Digits in Geocodes	-90.0 to 90.0 for latitude, -180.0 to 180.0 for longitude	Degrees, decimal or degrees, minutes and seconds with decimals	22 (including decimals , a space, and 2 sig. figs. for DMS)
Succinctness	Number of ASCII characters per point with full geocode	Greater than or equal to zero	ASCII characters	27 (includes EOL)
Succinctness	Coordinate Digit Density Function	Graph (one value per significant digit)	None	NA
Succinctness	S (entropy measure based on CDDF)	0-1.8	None	NA
Succinctness	Number of Algorithm Steps for retrieval	Greater than one	Steps	NA
Succinctness	Number of computations to ASCII conversion	Greater than one	Computations	NA
Succinctness	Number of look-ups performed	Greater than or equal to zero	Look-ups	NA
Succinctness	CPU time for conversion	Greater than zero	seconds	NA
Succinctness	User encoding and decoding time	Greater than zero	seconds	Needs human subject tests
Definitiveness	Overlap as a proportion of total space covered by grid	Greater than or equal to zero	Ratio	0.0 for latitude approx. 0.001 for longitude
Definitiveness	Weighted Overlap as a proportion of total space covered by grid	Greater than or equal to zero	Relative value, reflecting multiple counts	0.0 for latitude Infinity at 90N and 90S, 0.0 elsewhere
Definitiveness	Similarity coefficient for adjacent cells/points	Ratio of bitwise mismatch to mismatch + match (0.0-1.0)	Ratio	At 2 sig. fig for seconds, 0.9091
Exhaustiveness	Range of resolutions covered	Two values, both representative fractions or ground distances	Ratios	1:400,000,000 to 1:1
Exhaustiveness	Range of precision	Significant Digits or parts per million	Digits/PPM	5 (whole degrees) to 17
Exhaustiveness	Proportion of earth covered at finest resolution and precision	Greater than zero	Ratio	1.0

Exhaustiveness	Compared to Geographic, pixel loss and duplication ratios	Loss 0.0-1.0 Duplication greater than zero	Ratio	NA (comparison base)
Exhaustiveness	Ratio of atom to resolution	Greater than zero	Ratio (smallest desired resolution/precision)	1m/0.31 m = 3.23
Hierarchy	Number of reprojections within system	Greater than or equal to zero	Different projections/central meridians, points of tangency or secancy	0
Hierarchy	Number and length of joints	Greater than or equal to zero	Count	0
Hierarchy	Number of recursions from base to atom level	Greater than one	Recursion levels	3 (for DMS)
Uniqueness	Average number of significant digits in coordinates that distinguish between adjacent cells or points	Greater than one	Digits	1
Uniqueness	Coordinate Digit Density Function for point pairs	Graph	NA	NA
Uniqueness	Match ratio for adjacent coordinates	1-(mismatch/(match + mismatch)) 0.0-1.0	None	At 2 sig. fig for seconds, 0.9091
Intuitive Understanding	Number of facts that explain system	Greater than one	Facts	8
Intuitive Understanding	Number of parameters externally defined (magic numbers)	Greater than zero	parameters	4
Intuitive Understanding	Length of explanation/documentation	Greater than zero	Words (Source: Snyder Map Projections: A Working Manual)	C. 1000

Intuitive Understanding	Time to achieve error free use	Greater than zero	days, minutes	NA
Intuitive Understanding	Educational level required	K-16	Grade Level	10
Intuitive Understanding	Time to achieve explanation	Greater than one	minutes	20
Intuitive Understanding	Frequency of retraining	Greater than zero	months	NA
Intuitive Understanding	Memory recall of common geocodes	Binary, or Human subjects derived error rate	Yes/No or proportion of error	NA
Tractable	Availability of Method	Formulae or algorithm in literature/web	Yes/No	Yes
Tractable	Size of computer program for use	Smallest available computer program	Bytes	NA
Tractable	Steps in program logic	Lines of code	Lines	NA
Tractable	Program execution time	CPU or user time	seconds/point	NA
Tractable	Supports sampling and generalization	Boolean	Yes/No	Yes
Accurate	Test against independent source of higher authority	1-(mismatch/(match + mismatch)) 0.0-1.0	None	1.0 (self)
Accurate	Forward to inverse transformation comparison	1-(mismatch/(match + mismatch)) 0.0-1.0	None	1.0 (self)
Accurate	Repeatability	Proportion in error	None	1.0 (assumed)
Accurate	RMS or other single accuracy value for whole data set, as projected	distance or standard deviation	meters	0.31 m

APPLICATION

The criteria listed above, coupled with the metrics in Table 3, are a foundation around which objective comparisons between global grid systems is possible. The illustrative values entered in the table for the geographic coordinate system are provided as a first set of estimates, and will be refined over time. Similar values for the various grid systems in use can be computed accordingly. Comparison can then take the form of a series of greater than tests, by principal components analysis of the scores, or by the computation of weighted aggregate scores. No such comparison is attempted here, but research is invited in this new and potentially useful comparative approach to the analysis of global grids.

CONCLUSION

It is foolish to believe that a single grid system would ever serve the needs of all users. Nevertheless, for particular applications and disciplines, placement and contrasting of systems within the proposed framework would allow objective decisions to be made about which grid to select. An advantage of the analytical approach to grid selection is that the comparative metrics point out both strengths and weaknesses of any system for a particular application.

Analytical cartography can serve in comparing global grids to conduct meta-analysis of entire systems. The metrics and methods proposed can serve to illustrate possible enhancements, improvements and modifications to existing grid systems that may be of considerable benefit to map producers and users alike. Whatever the outcome, the current era of Internet and World Wide Web based cartography will ensure that the user, rather than the cartographer, surveyor, or geodesist will increasingly influence the future of the mapping sciences. Occam's razor has been proposed as applicable to global grid systems, that is, given two equally useful and powerful grid systems, the better one is the simpler of the two. User demand assessment and user testing are only now becoming regular tools in the cartographer's toolbox. Web mapping both demands immediate solutions to the inadequacies of particular grid systems and provides a somewhat objective means by which user testing can be conducted rapidly and in sufficient numbers to move beyond the currently favored "30 geography students" that human subjects tests of mapping applications tend to use.

The future, quite clearly, will reward those systems that meet their Web searching demands with de facto acceptance and therefore authority. For over a century, cartography has allowed the competitive coexistence of global grid systems devised for different applications. As surveying and mapping yield to mobile mapping applications such as navigation and high-precision positioning, it is hoped that the methods and metrics proposed here can lead to some effective choices for the future based on analysis and quantitative methods rather than

subjectivity and bias.

REFERENCES

Clarke, K. C. (1995) *Analytical and Computer Cartography*. Englewood Cliffs, NJ: Prentice Hall.

Clarke, K. C. and P. D. Teague (1998) "Cartographic Symbolization of Uncertainty" Proceedings, ACSM Annual Conference, March 2-4, Baltimore, MD. (CD-ROM)

Dretske, F. I. (1999) *Knowledge and the Flow of Information.* Stanford University: CSLI Publications.

Dutton, G. H. (1999) *A hierarchical coordinate system for geoprocessing and cartography*. Berlin ; New York: Springer.

Goodchild, M. F. (1994) "Criteria for evaluation of global grid models for environmental monitoring and analysis", Handout from NCGIA Initiative 15, see Spatial Analysis on the Sphere: A Review, by Rob Raskin, NCGIA Technical Report 94-7. Copy courtesy of Waldo Tobler.

Goodchild, M. F. and Gopal, S. eds. (1989) *Accuracy of Spatial Databases*. London: Taylor and Francis.

Howse, D. (1997) *Greenwich Time and the Longitude*, London: Philip Wilson.

Kimerling, A. J., K. Sah, D. White and L. Song (1999) "Comparing Geometrical Properties of Global Grids", *Cartography and Geographic Information Systems*, vol. 26, no. 4, pp. 271-88.

Monmonier, M. S. (1995) *Drawing the line : tales of maps and cartocontroversy*, New York :H. Holt.

Mulcahy, K. A. (1999) *Spatial Data Sets and Map Projects: An Analysis of Distortion*. Ph.D. Dissertation, City University of New York: University Microfilms: Ann Arbor, MI.

Mulcahy, K. A. and K. C. Clarke. "Cartographic Visu alizations of Map Projection Distortion: A Review", Car tography and Geographic Information Science, vol 28, no. 3, pp. 167-181.

Robinson, A. H. et al. (1995) *Elements of Cartography*. New York: J Wiley. 6th. ed.

Shannon, C. (1948) "The Mathematical Theory of Communication", *Bell System Technical Journal*, July/October.

Tobler, W. 1970. "A Computer Movie Simulating Urban Growth in the Detroit Region." *Economic Geography* 46(2):234-240.

Tobler, W. R. and Z. Chen (1986) "A Quadtree for Global Information Storage", *Geographical Analysis,*vol. 18, no. 4, pp. 360-371.

Tobler, W. R. (2000) "The development of Analytical Cartography: A Personal Note," *Cartography and Geographic Information Systems*, Vol. 27, No. 3, pp. 189-194..

Tukey, J. W. (1977) *Exploratory Data Analysis.* Reading, MA: Addison-Wesley.