Available online at www.sciencedirect.com

ScienceDirect

Computers, Environment and Urban Systems 32 (2008) 174–187

Computers, Environment and Urban Systems

www.elsevier.com/locate/compenvurbsys

Location coding on icosahedral aperture 3 hexagon discrete global grids

Kevin Sahr *

Department of Computer Science, Southern Oregon University, 1250 Siskiyou Boulevard, Ashland, OR 97520, United States

Abstract

Discrete global grid systems (DGGSs) represent a relatively new, but increasingly popular, approach to the problem of representing geospatial location on computer systems. Despite growing interest amongst potential users in icosahedral aperture 3 hexagon DGGSs, the practical use of such systems has been hindered by a lack of efficient spatial indexing methods. In this paper we discuss the two primary approaches to developing multi-resolution location coding systems for DGGSs: pyramid addressing and path addressing. We then describe an efficient pyramid addressing system for icosahedral aperture 3 hexagon DGGSs, the quadrilateral 2-dimensional integer system. After reviewing the problems inherent in developing path addressing systems for hexagon-based DGGSs we describe a class of path-based location coding solutions for icosahedral aperture 3 hexagon DGGSs called modified generalized balanced ternary, and show how this system can be used to index vector data. We then discuss a subset of this system, the icosahedral aperture 3 hexagon tree, which can be used to index raster and bucket data structures. Conversion algorithms to/from geodetic coordinates are discussed.

1. Introduction

Discrete Global Grid Systems (DGGSs) (Sahr, White, & Kimerling, 2003a) are a relatively new approach to the representation of location on the earth's surface, based on multi-resolution partitions of spherical polyhedra. DGGSs have been proposed based on cells that are triangles, squares, diamonds, and hexagons.

The cells of any DGGS can form the basis of at least three types of geospatial data structure. Under the most common usage the DGGS cell regions constitute the pixels of a raster system, with data values assigned to the areal cell regions. Additionally, as proposed by Dutton (1989), the center points of DGGS cells can form a multi-resolution vector system. In this case each zero-dimensional point on the earth's surface is mapped to the DGG center point of the cell region in which it occurs, at each resolution of the DGGS. Finally, the DGGS cell regions can serve as buckets into which data objects are assigned based on their location. Depending on the application, a data object can either be assigned to the finest resolution cell that entirely contains it, or to the coarsest resolution cell which uniquely distinguishes it from all other data objects in the data set of interest. The DGGS then forms the basis for a spatial database that can be efficiently queried due to the regular geometry and hierarchical structure of the DGGS.

In order to be usable as a data structure, each cell in a DGGS must have assigned to it one or more unique *location codes*, which together constitute a *location coding system* — a particular computer representation of geospatial location. On contemporary computer systems each of these location codes consists of a string of bits. A quantization operator must be defined which maps each location on the earth's surface to a single location code or set of location codes; usually the inverse mapping is also defined.

Many DGGS alternatives have been known to the spatial data structures community for over a decade (see, for example, Samet's (1989, 1990) classic two-volume survey of spatial data structures). The data structures community has focused almost exclusively on DGGSs based on the recursive 4-fold decomposition of triangular cells. Such grids induce forests of quadtrees that can be location coded

* Tel.: +1 541 552 6978; fax: +1 541 552 6171.
E-mail address: sahrk@sou.edu

0198-9715/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compenvurbsys.2007.11.005

using traditional quadtree approaches, as do DGGSs based on the 4-fold decomposition of square or diamond cells.

However, a great deal of recent attention among end-users and GIS researchers has focused on DGGSs based on cells that are hexagons, which have many advantages as a basis for constructing discrete grids. Among the three regular polygons that tile the plane, hexagons are the most compact, they quantize the plane with the smallest average error (Conway & Sloane, 1998), and they provide the greatest angular resolution (Golay, 1969). Unlike grids based on square or triangle cells, hexagonal cells exhibit uniform adjacency. That is, each hexagon cell has six neighbors, all of which share an edge with it, and all of which have centers exactly the same distance away from its center. Each hexagon cell has no neighbors with which it shares only a vertex, as do square or triangular cells. This fact alone has made hexagons increasingly popular as bases for discrete spatial simulations. Frisch, Hasslacher, and Pomeau (1986) argue that the six discrete velocity vectors of the hexagonal lattice are necessary and sufficient to simulate continuous, isotropic, fluid flow. A recent textbook (Rothman & Zaleski, 1997) on fluid flow cellular automata is based entirely on hexagonal meshes, with discussions of square grids included "only for pedagogical calculations" (triangle grids, which are even more limited for this purpose, are not mentioned). Studies by GIS researchers (Kimerling, Sahr, White, & Song, 1999) and mathematicians (Saff & Kuijlaars, 1997) indicate that hexagon-based DGGSs may have clear advantages for many applications. A hexagon-based grid has been used by the US EPA for global sampling problems (White et al., 1992). And hexagon-based Geodesic DGGSs have been proposed at least four times in the atmospheric modeling literature (Williamson, 1968; Sadourny, Arakawa, & Mintz, 1968; Heikes & Randall, 1995a, 1995b; Thuburn, 1997) — to our knowledge more often than any other DGGS topology.

But unlike DGGSs based on triangles, squares, or diamonds, multi-resolution hexagon DGGSs cannot be location coded using a traditional quadtree approach. Multiple resolutions of hexagon cells cannot be created by simple aggregation of atomic pixels, nor by recursive partition. Such grids do not naturally induce hierarchical data structures that are quadtrees, or even strictly trees. Yet hexagon-based DGGSs are clearly an important approach that cannot be ignored by the geospatial data structures community. To quote the conclusion of Sahr et al. (2003a): "a significant effort must be made by the data structures community to develop and evaluate algorithms for the regular, but non-tree, hierarchies they form." In particular, we believe that the development of reasonable hierarchical location coding schemes is basic to such an effort.

It is possible to construct multi-resolution hexagon grids that exhibit regularities that can be used to develop hierarchical location coding systems. Multi-resolution hexagon grids can be formed such that the center point of each resolution k cell is also a cell center point in resolution $k + 1$. Such a hierarchy is known as a *central place* (Christaller, 1966) or *aligned* (Sahr et al., 2003a) hierarchy. Dacey (1965) notes that aligned hexagon grids can be formed for any aperture $h^2 + hk + k^2$, where h and k are any positive integers. DGGSs have been proposed using hexagon hierarchies in which increasing resolution reduces cell area by a factor of 3 ($h = 1, k = 1$) or 4 ($h = 0, k = 2$). These are known as *aperture 3* and *aperture 4* grids, respectively.

Figs. 1 and 2 illustrate aperture 3 and 4 hexagon hierarchies defined symmetrically on a single triangle. Twenty such triangles can be tiled onto the triangular faces of a spherical icosahedron to form a DGGS. It should be noted that the sphere cannot be totally tiled using just hexagons; the cells of a hexagon-based icosahedral DGGS always include exactly twelve pentagonal cells at each resolution, centered on the twelve vertices of the icosahedron. Proposed hexagon DGGSs include the icosahedral Snyder equal area aperture 3 hexagon (ISEA3H) DGGS (Sahr et al., 2003a), which has the advantage of equal-area hexagonal cells. Fig. 3 illustrates three resolutions of the ISEA3H.

As previously noted, in order for hexagon DGGSs such as the ISEA3H to be useful as data structures they must

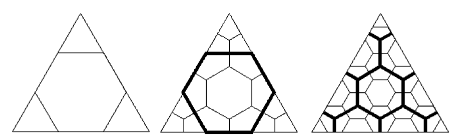

Fig. 1. Three resolutions of an aperture 3 hexagon hierarchy defined symmetrically on a single triangle.

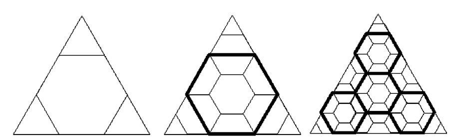

Fig. 2. Three resolutions of an aperture 4 hexagon hierarchy defined symmetrically on a single triangle.

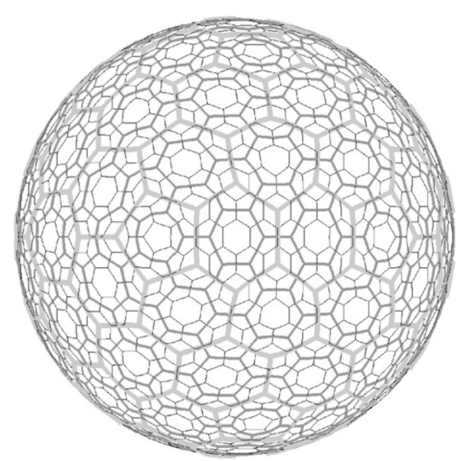

Fig. 3. Three resolutions of the ISEA3H DGGS.

have location coding systems defined on them. Multi-resolution location coding systems generally take one of two basic forms: pyramid addressing and path addressing. Our task in this paper, then, is to propose location coding systems for icosahedral aperture 3 hexagon DGGSs (such as the ISEA3H) using each of these two approaches. In the next section we will introduce the quadrilateral 2di pyramid addressing system, which was implemented in the DGGRID software system (Sahr, 2002) but which has never been previously described for publication. We will then discuss the advantages of path addressing systems, and the difficulties inherent in developing such systems for hexagon-based DGGSs. We then introduce a class of path-based location coding solutions for icosahedral aperture 3 DGGSs called icosahedral modified generalized balanced ternary, and describe how this system can be used to index vector data. Next we describe a subset of this system that can be used to index raster and bucket data structures, the icosahedral aperture 3 hexagon tree. Finally, we describe a quantization algorithm for these path-based addressing systems, which enables their use as geospatial location coding systems.

2. Pyramid addressing on aperture 3 hexagon grids: the quadrilateral two-dimensional integer system

The pyramid addressing approach (Burt, 1980) constitutes one of the two principle approaches to developing multi-resolution location coding systems. Under this approach each DGGS cell is assigned a unique location code within the DGG of corresponding resolution. This single-resolution location code may be multi-dimensional or linear. Given a resolution k cell with single-resolution location code *k-address*, we can designate the DGGS pyramid location code to be the two-tuple *(k, k-address)*. A multi-resolution DGGS address representation of a location can be constructed by taking the series of single-resolution DGG addresses for that location, ordered by increasing resolution.

Pyramid addresses are useful in applications that work with complete single resolution data sets. Various pyramid address location coding systems with associated quantization operators are known for grids based on triangles, squares, diamonds, and hexagons.

Traditional planar cartesian coordinate systems employ two coordinate axes that are perpendicular to each other. Hexagon systems, however, naturally form three axes that are at 60° angles to each other. As illustrated in Fig. 4, there are two natural orientations of these three axes relative to the traditional cartesian coordinate system, which we designate *Class I* and *Class II* as a generalization of terminology developed for triangle grid systems by Fuller (1975).

Two of these axes are sufficient to uniquely identify each hexagon. Two-axis planar hexagon coordinate systems have been used in the development of a number of algorithms for hexagon grids, including quantization to/from

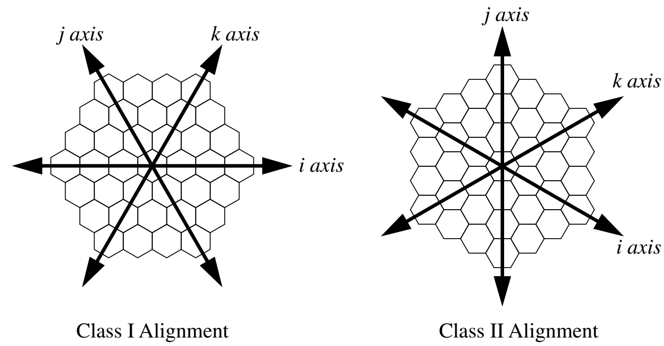

Fig. 4. Three-axis hexagon coordinate systems.

cartesian coordinates (van Roessel, 1988), metric distance (Luczak & Rosenfeld, 1976), vector addition and subtraction (Snyder, Qi, & Sander, 1999), neighbor identification (Snyder et al., 1999), adapted Bresenham's line and circle rasterization (Wuthrich & Stucki, 1991), edge detection (Abu-Bakar & Green, 1996), line-of-sight (Verbrugge, 1997), field-of-view (Verbrugge, 1997), image gradient determination (Snyder et al., 1999), and variable conductance diffusion (Snyder et al., 1999).

Note that, since a quantization operator is defined for 2-axis systems, any alternative location coding for a hexagon grid trivially implements the other algorithms listed provided a mapping is defined between the alternative system and a 2-axis system. Location codes in the alternative coding would be converted to 2-axis coordinates, and the algorithm would then be applied using that system. Any resulting location codes would be converted back to the alternative location coding.

We choose the i and j axes as a coordinate system basis because they are most useful in constructing pyramid addresses for icosahedral DGGSs. We designate the resulting coordinate system a *two-dimensional integer* (*2di*) coordinate system. Fig. 5 illustrates the assignment of coordinate addresses in a Class I 2di coordinate system.

A planar multi-resolution aperture 3 hexagon grid system may be formed as follows. Begin with a single resolution Class I hexagon grid and call it the resolution k grid. To form the next finer grid (resolution $k + 1$), create a Class II hexagon grid consisting of hexes with exactly 1/3 the area of the resolution k hexagons, with the resolution $k + 1$ hexagons centered on the vertices and center points of the resolution k hexagons. Repeat this process the desired number of resolutions, alternating between Class I and Class II grids at each successive resolution. Fig. 6 illustrates four resolutions of such a grid. (Note that the series may also be started with a Class II grid, again with successive resolutions alternating class).

A unique pyramid address of the form $[r, (i, j)]$ may be assigned to each hexagon in a planar multi-resolution grid, where r is the resolution of the hexagon and (i, j) is the 2di address of the hexagon on the resolution r grid.

The twenty triangular faces of the icosahedron can be paired into ten quadrilaterals as illustrated in Fig. 7. Each

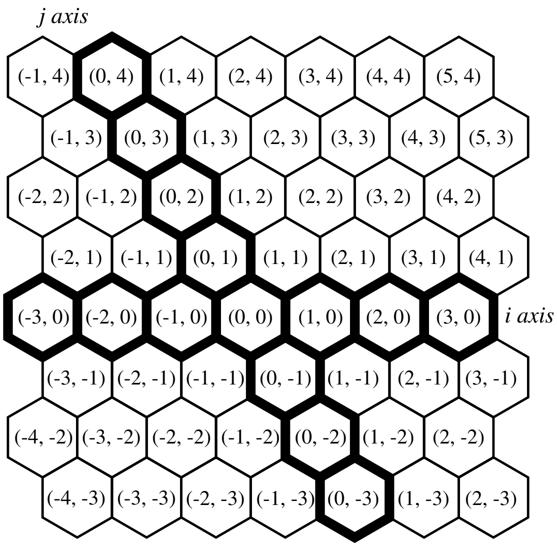

Fig. 5. Class I 2di coordinate system.

of these quadrilaterals can be addressed using a Class I 2di coordinate system, with each 2di coordinate system origin corresponding to one of the pentagonal vertex cells and with the Class I axes lying along the quadrilateral edges. Two pentagonal vertex cells are left-over and can be treated as single-cell 2di coordinate systems at every resolution. Fig. 8 shows 2di coordinate systems on an icosahedron unfolded onto the plane and with the quadrilaterals given one possible numbering. The pentagonal vertex cells have been drawn as hexagons.

Class II grids do not conveniently align with the Class I coordinate axes naturally defined by the quadrilateral edges. One solution to this problem can be found by noting that for every cell of a Class II resolution k grid there is a single Class I resolution $k + 1$ cell centered on that resolution k cell. Thus without ambiguity we can assign to each resolution k cell the coordinates of the Class I resolution $k + 1$ cell centered upon it. Fig. 9 illustrates one Class II resolution quadrilateral addressed using this method.

A unique pyramid address of the form$\{r, [q, (i,j)]\}$ may be assigned to each cell in the multi-resolution DGGS, where r is the resolution of the hexagon, q is the quadrilateral on which the cell occurs, and (i, j) is the 2di address of the hexagon on the quadrilateral q resolution r grid. This icosahedral coordinate system is designated the *quadrilateral 2di* (*q2di*) system (Sahr, 2002).

3. Path addressing on aperture 3 hexagon grids

The second major approach to location coding multi-resolution grids is the path addressing approach. In this section we will discuss the advantages of path addressing systems over pyramid addressing systems, and the problems inherent in developing path addressing systems for hexagon-based DGGSs.

The path addressing approach to location coding takes advantage of the fact that the resolution k quantization of a geospatial location into a DGGS cell restricts the possible resolution $k + 1$ quantization cells to those whose regions overlap or are contained within the resolution k cell. We can construct a *spatial hierarchy* by designating that each resolution k cell has as children all resolution $k + 1$ cells whose regions overlap or are contained within its region. A *path address* is a location code that specifies the path through a spatial hierarchy that corresponds to a multi-resolution location quantization. Path address location codes are often linear, consisting of a string of digits where each digit in the code corresponds to a single resolution and specifies a particular child cell of the parent cell specified by the address prefix. If the numerical base of the digits is equal to the number of children at each resolution, then each possible digit value can uniquely and optimally indicate a particular child cell; in this case the path address is known as a *trie* (Fredkin, 1960).

Since path addresses do not store redundant multi-resolution location information, they can be much more compact than multi-resolution pyramid addresses. And because path address location codes can be constructed such that the number of digits in the code corresponds to the maximum resolution, or precision, of the address, path addresses automatically encode their precision (Dutton, 1989), obviating the need for separate precision metadata.

Path addresses also often simplify the development of hierarchical algorithms. In particular, the coarser cell represented by the prefix of a location code can be used as a coarse filter for the proximity operations containment, equality, intersection/overlap, adjacency, and metric distance. These proximity operations form the primary spatial queries used on spatial databases, and such queries are thus rendered more efficient by the use of path addresses. For example, take a bucket system where data object boundaries are assigned to the smallest containing cell. A query may ask for all data objects whose locations intersect a particular region. We can immediately discard from consideration all objects in bucket cells that do not intersect the smallest containing bucket cell of the query region (Samet, 1989).

Path addresses are traditionally associated with *congruent* grids, where the regions associated with resolution k cells are the union of resolution $k + 1$ cells (Sahr et al., 2003a). Congruent grid systems have spatial hierarchies that form traditional trees, where each child has one and only one parent. The most commonly used such structure is the square quadtree, consisting of a square root cell that is sub-divided into four square children, each of which is recursively sub-divided into four children, and so forth to the desired resolution. At each successive resolution one of four digits (usually 0, 1, 2, or 3) is concatenated to the location code of the parent cell to indicate the location code of each child cell. Quadtree-like hierarchies can also be created by the fourfold recursive subdivision or aggregation of triangle or diamond cells. Thus DGGSs based on the fourfold recursive subdivision of squares, triangles, or diamonds can all be location coded as forests of quadtrees.

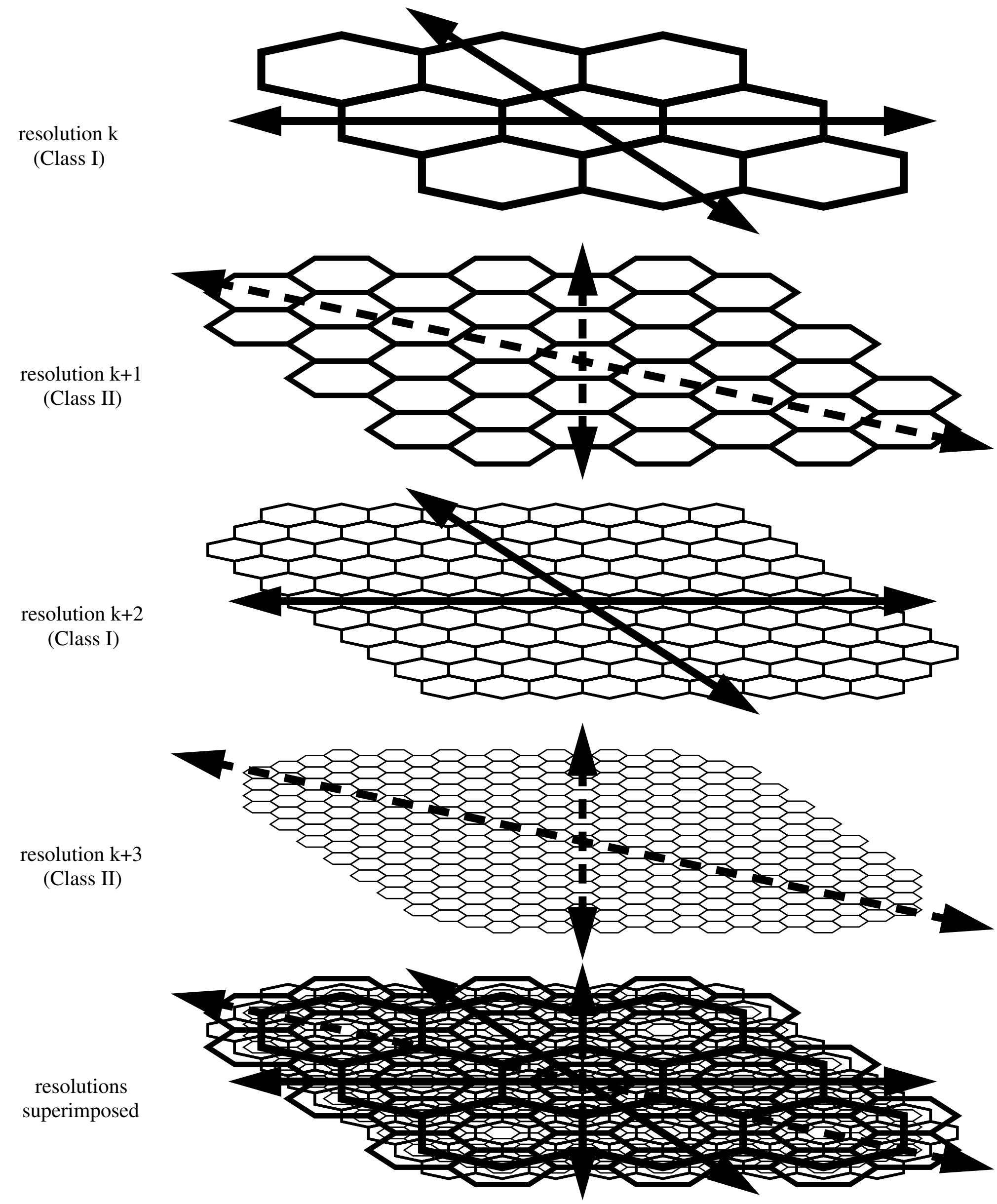

Fig. 6. Four resolutions of an aperture 3 hexagon grid system.

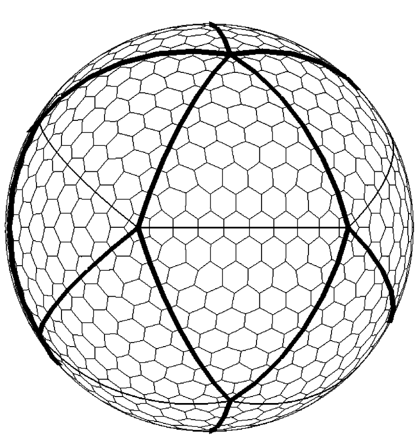

Fig. 7. Icosahedral faces paired to form ten quadrilaterals.

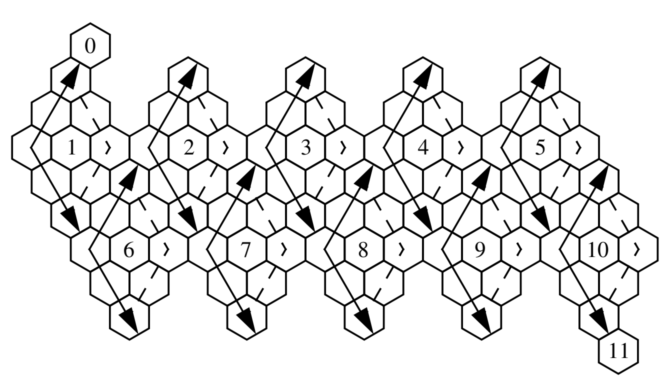

Fig. 8. Unfolded icosahedron with 2di coordinate systems. Note that the pentagonal icosahedron vertex cells are rendered as hexagons.

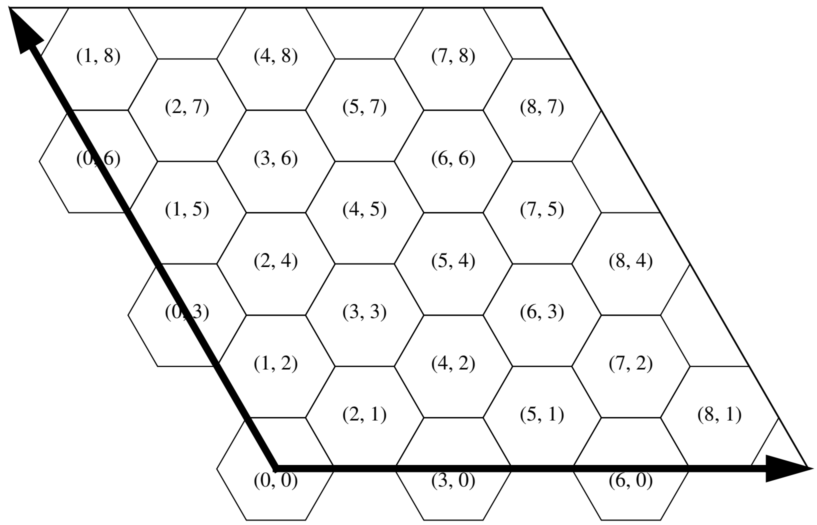

Fig. 9. Class II grid on a single quadrilateral addressed using underlying Class I coordinates.

Hexagon DGGSs are necessarily incongruent; it is impossible to exactly decompose a hexagon into smaller hexagons, or, conversely, to aggregate small hexagons to form a larger one. For example, as previously noted Fig. 1 illustrates three resolutions of an aperture 3 hexagon hierarchy. Note that each cell can have up to three parents, and that the spatial hierarchy induced by an aperture 3 hexagon grid is therefore not a tree. Because hexagons do not form the congruent hierarchies normally used to construct path addressing systems, hexagon-based DGGSs have traditionally been addressed using pyramid addressing systems.

Note that it is possible to aggregate hexagons to form hierarchical addressing systems, though the aggregated hexagon units are not themselves hexagons. In particular, hexagons can be aggregated in groups of seven to form coarser resolution objects which are almost hexagons, and these groups can be aggregated again into pseudo hexagons of even coarser resolution, and so on, as illustrated in Fig. 10. The most widely-used addressing of each unit of seven hexagons (or pseudo hexagons) is a generalization of one-dimensional Balanced Ternary addressing (Knuth, 1998), and the system is therefore named Generalized Balanced Ternary (GBT) (Gibson & Lucas, 1982). Balanced Ternary uses three-valued digits that represent -1, 0, or 1. As illustrated in Fig. 11, GBT generalizes this notation to the three axes of a hexagon grid. In any seven-hex unit the central hex is designated digit 0, while the remaining digits 1 through 6 are arranged so that digits on opposite sides of the central hex are binary complements of each other. This elegant location coding has become the most widely used path addressing system for location coding planar hexagon grid systems. And while it does not directly meet our needs in location coding hexagon-based DGGSs, as we will see in the next section it does provide the basis for a class of solutions.

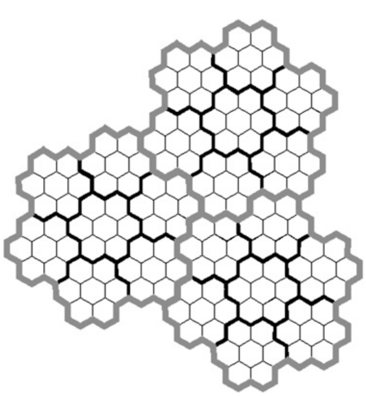

Fig. 10. Sevenfold hexagon aggregation into coarser pseudo-hexagons.

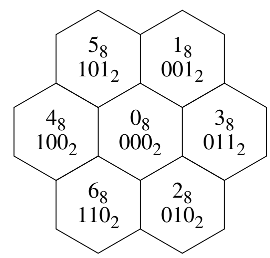

Fig. 11. Arrangement of digits in a generalized balanced ternary aggregation unit. Note that hexagons on opposite sides of the central hexagon have digits that are binary complements.

4. Path address location coding for aperture 3 hexagon DGGSs: icosahedral modified generalized balanced ternary

In this section we introduce a class of path-based location coding solutions for icosahedral aperture 3 DGGSs called Icosahedral Modified Generalized Balanced Ternary, and show how this system can be used to index vector data.

Let (x, y) be a zero-dimensional point location on the plane. Then a resolution k aperture 3 hexagon grid quantization of this point restricts the possible resolution $k + 1$ quantization of the point to the seven resolution $k + 1$ hexagons that overlap the resolution k hexagon. This is illustrated in Fig. 12.

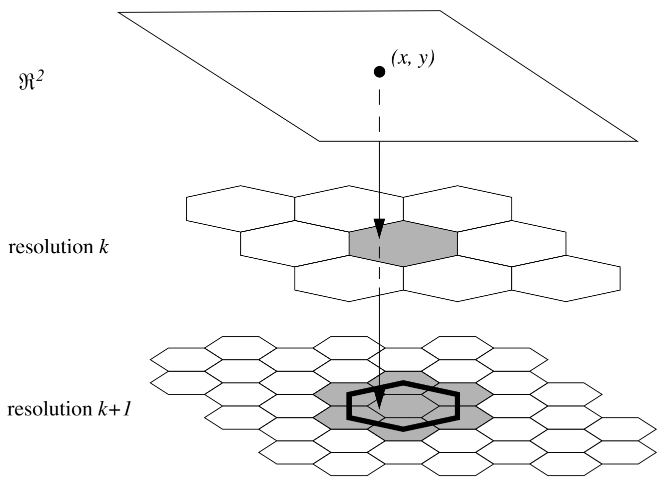

Fig. 12. Quantization of a point at one resolution restricts possible quantization candidates at higher resolutions.

Given a linear code for a resolution k cell in a planar aperture 3 hexagon grid, a multi-resolution hierarchical location coding scheme can be specified by assigning specific digits to each of the seven possible resolution $k + 1$ cells, and concatenating to the resolution k code the digit corresponding to the child cell in which the point lies. This scheme can be applied iteratively until the desired maximum resolution is achieved.

We note that, as illustrated in Fig. 12, the children of a cell in an aperture 3 hexagon spatial hierarchy correspond to one GBT unit. Assigning digits using a GBT-based approach yields the hierarchical location coding system Modified GBT (MGBT). Figs. 13 and 14 show the assignment of resolution $k + 1$ address digits based on Class I/Class II resolution k addresses, respectively.

Each resolution $k + 1$ hexagon may overlap up to three different resolution k parents to which different regions of it belong. This means that MGBT effectively addresses subregions of hexagons, such that each entire hexagon is not assigned a unique location code, and the resolution k quantification of a point (x, y) effectively assigns the point to the

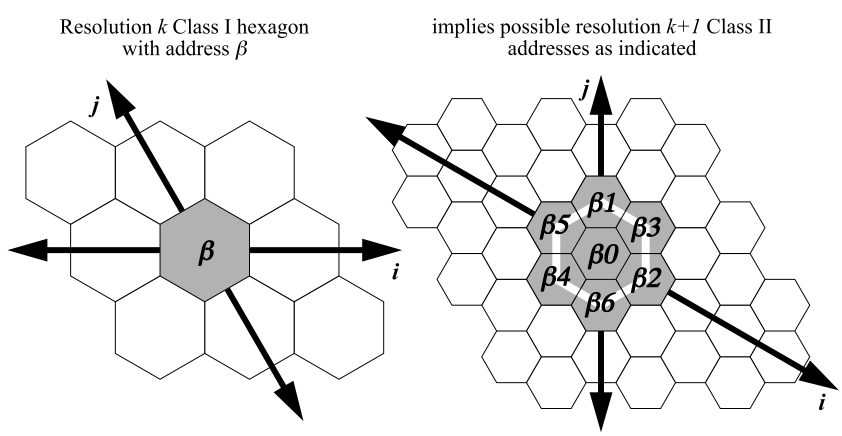

Fig. 13. MGBT addressing of Class II children.

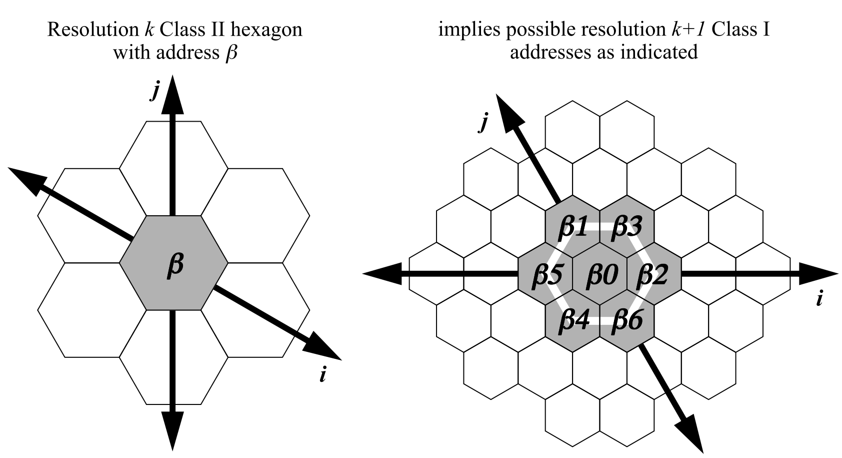

Fig. 14. MGBT addressing of Class I children.

intersection of the cells that contain it at resolutions *0* through k. Fig. 15 illustrates the sub-regions that are addressed at the next finer resolution by three Class I hexagons. Fig. 16 illustrates the location coding of a point in a 3-resolution MGBT system.

Since at each resolution one of seven digits is added to the location code, it is possible for each digit to be represented using 3 bits. Note that 3 bits can represent eight distinct digits. The decimal digits 0–6 have been assigned as illustrated in Figs. 13 and 14 above to correspond with the GBT system. The remaining eighth digit, decimal 7 or binary 111, can serve a number of useful functions. If the addresses are variable length, then a 7-digit can be concatenated to an address to indicate address termination. If the addresses are fixed length, a 7-digit can be used to indicate that the remaining higher resolution digits are all center digits (i.e. zero), and therefore that additional resolution will not add information to the location.

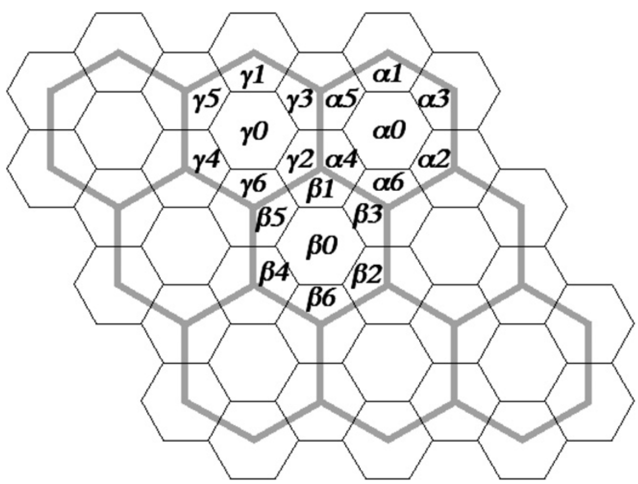

Fig. 15. Sub-regions addressable at next finer resolution by three Class I MGBT cells.

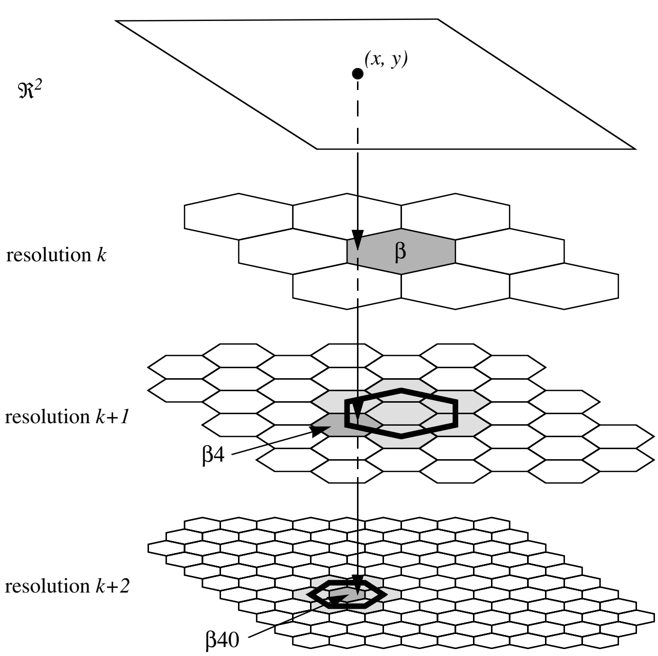

Fig. 16. The point (x, y) location coded using a 3-resolution MGBT system, where (x, y) lies in the resolution k cell with address β.

Note that the MGBT codes for point locations location coded using MGBT can be truncated such that any prefix of the code yields a valid quantification of the point location at a coarser grid resolution. This allows prefixes of the code to be used as a coarse filter for the point proximity operations equality, adjacency, and metric distance.

We can extend planar MGBT to a DGGS location coding system by tiling an icosahedral aperture 3 hexagon DGGS with MGBT tiles. Tiles centered on the twelve icosahedral vertices form pentagons and thus require special tiling units. These can be constructed by deleting one-seventh of the sub-hierarchy generated in the hexagon case. This can be accomplished by following the procedure outlined for hexagonal MGBT tiles but with a single sub-digit sequence deleted. That is, for a pentagonal tile with base address A, all sub-cells are indexed as per the corresponding MGBT indexing except that sub-cells with sub-indexes of the form AZd are not generated, where Z is a string of 0 or more zeroes and d is the sub-digit sequence (1, 2, 3, 4, 5, or 6) chosen for deletion. All hierarchical descendants of these deleted sub-cells are likewise not indexed. We use *MGBT-d* to indicate an MGBT tile with sub-digit sequence d deleted (e.g., MGBT-1 would indicate an MGBT tile with sub-digit sequence 1 chosen for deletion). Fig. 17 illustrates the addresses assigned to the sub-regions of such a pentagonal cell.

Given hexagonal and pentagonal MGBT tiles, we may tile the icosahedron to create the icosahedral MGBT (iMGBT) location coding system. In order to fully specify a geospatial coding system based on the iMGBT, a fixed orientation must be specified for each tile. One approach to achieving this is to unfold the icosahedron onto the plane and then specify that each tile be oriented consistently with this planer tiling. Fig. 18 shows one such orientation. Each base tile is labeled with a base tile number.

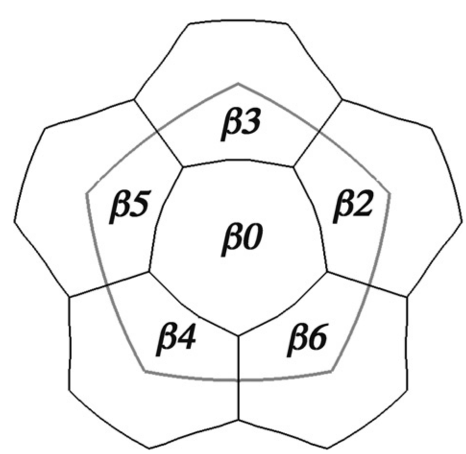

Fig. 17. Addresses of the sub-regions of an MGBT-1 pentagonal tile with address β.

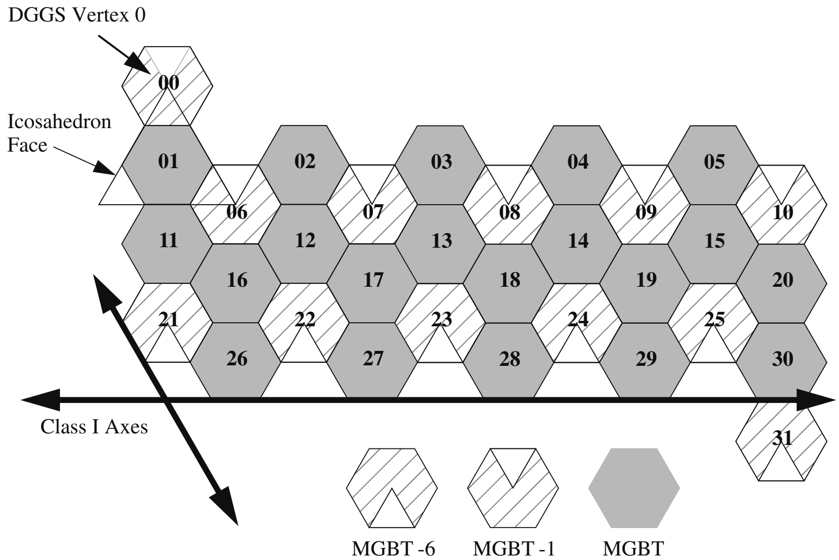

Fig. 18. iMGBT base tiling on an unfolded icosahedron.

An iMGBT location code can have one of the following four easily inter-convertible forms:

1. *Character string code form*. The location code consists of a string of digits beginning with the two-digit base tile code (00, 01, 02, ..., 31) followed by the digit string corresponding to the appropriate address of each finer resolution within the tile.
2. *Integer code form*. The character string code form may be interpreted and stored as a single integer value.
3. *Modified integer code form*. When displaying integer values leading zeroes are usually removed. This will result in differences in the number of digits between codes on base tiles 00–09 and those of the same resolution but on other base tiles. This can be remedied by adding the value of 40 (four being the lowest unused value for the 10's digit) to each of the base tile values so that they are numbered 40, 41, ...,71. Note that since digits 4–7 are not used as leading digits in the integer code form, modified integer codes can be unambiguously distinguished from the standard integer codes.
4. *Packed code form*. Under this form, the base tile codes are stored as five-digit binary numbers. Sub-codes within each tile are stored as a packed series of three-bit binary digits (as described above) and appended to the base tile number to fully specify a geospatial code. Under this scheme, codes up to resolution 10 can be stored in 32 bits of contiguous storage, and codes up to resolution 20 can be stored in 64 bits of contiguous storage.

5. Raster and bucket location coding: the icosahedral aperture 3 hexagon tree

As mentioned above, the iMGBT effectively addresses sub-regions of cells and thus assigns multiple codes to many of the cells in an aperture 3 hexagon DGGS. This is reasonable if the DGGS is to be used as a vector system, since the primary data objects are zero-dimensional points that do indeed map to hexagon sub-regions. But in the case of raster or bucket systems data objects are mapped to entire cells and it is therefore useful to have a specific unique location code for each cell. If we wish to map the cells to physical resources — to perform parallel processing, for instance — the assignment of a unique code becomes a necessity.

For applications where unique cell location codes are required we must decompose the iMGBT into unique sub-trees with location codes that are a subset of the complete iMGBT addressing system. Here we will propose one possible decomposition, based on decomposing the planar MGBT hierarchy into a combination of two sub-tree topologies. We call this decomposition the planar *aperture 3 hexagon tree* (A3HT) (Sahr, Peterson, & Lutterodt, 2003b).

Each hexagon in an A3HT is assigned one of two generator hexagon types: *open* (*type A*), or *closed* (*type B*). An open generator at resolution k generates a single resolution $k+1$ hexagon that is a closed generator centered on itself. As illustrated in Fig. 19, a closed generator hexagon at resolution k also generates a single resolution $k+1$ closed generator hexagon at its center. But the closed generator in addition generates six resolution $k+1$ open generator hexagons, one centered at each of its six vertices.

An A3HT of arbitrary resolution can be created by beginning with a single open or closed hexagon and then recursively applying the above generator rules until the desired resolution is reached. Fig. 20 shows the first four resolutions of an A3HT generated by a resolution k closed generator hexagon.

Hexagons in an A3HT may be addressed using the corresponding MGBT codes. In all cases (Class I or Class II, open or closed generator), the address of centroid children

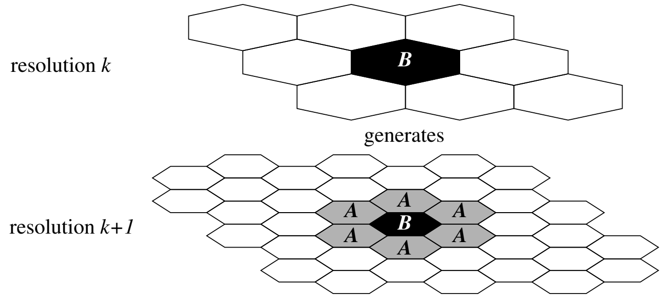

Fig. 19. A3HT closed (type B) generator hexagon.

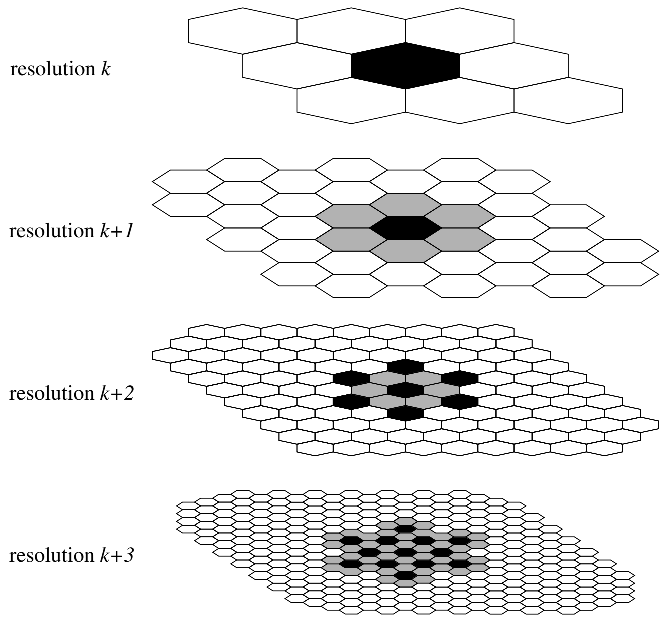

Fig. 20. Generation of four resolutions of an A3HT grid from a closed (type B) initial generator.

are formed by concatenating a zero digit with the parent hexagon address. The most widely-used addresses of vertex children of closed generators are formed by concatenating one of the digits 1–6 with the parent hexagon address, using the MGBT addressing given for Class I and Class II parents in Figs. 13 and 14, respectively. Fig. 21 illustrates the location coding of a cell in a 4-resolution A3HT system.

We can use the A3HT to address an icosahedral aperture 3 DGGS by tiling the DGGS with A3HT tiles. As in the case of the iMGBT, tiles centered on the twelve icosahedral vertices form pentagons and thus require special tiling units.

Aperture 3 pentagon tree (A3PT) tiling units are generated similarly to A3HT tiles except that one-seventh of the sub-hierarchy is deleted. Each pentagon in an A3PT is assigned one of two generator pentagon types: open (type A) or closed (type B). An open A3PT generator at resolution k generates a single resolution $k + 1$ pentagon that is a closed generator centered on itself. As illustrated in Fig. 22, a closed A3PT generator at resolution k also generates a single resolution $k + 1$ closed A3PT generator pentagon

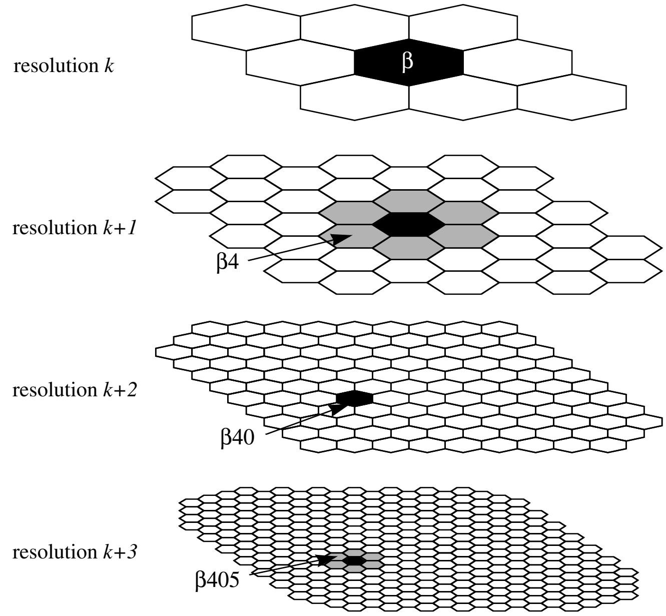

Fig. 21. Location coding a cell using a 4-resolution A3HT system.

at its center, but in addition it generates five resolution $k + 1$ open A3HT generator hexagons, one centered at each of its five vertices. An A3PT of arbitrary resolution can be created by beginning with a single open or closed A3PT and then recursively applying the above generator rules until the desired resolution is reached.

Like the MGBT-d tiles described in the previous section, A3PT tiles are addressed with a single sub-digit sequence deleted. *A3PT-d* is used to indicate an A3PT tile with the sub-digit sequence d deleted. This deletion occurs exactly as per the corresponding MGBT-d tile described in the previous section.

Given A3HT and A3PT tiles, we may tile the icosahedron to create the *icosahedral A3HT* (iA3HT). As with the iMGBT we can specify the orientation of base tiles by orienting them consistently with a planar unfolding of the icosahedron, as illustrated in Fig. 18. In this case we must also indicate which base tiles are closed and which are open; one consistent way to accomplish this is to use closed A3PT-6 and A3PT-1 tiles for the MGBT-6 and MGBT-1 tiles in Fig. 18, respectively, and open A3HT tiles for the hexagonal MGBT tiles. Fig. 23 illustrates the first six resolutions of an A3HT generation pattern on the ISEA3H DGGS. Note the pentagrams formed at each resolution by the A3PT tiles centered on the vertices of the icosahedron.

The iA3HT can be indexed using any of the four location code forms given for the iMGBT in the last section. In addition, all of the cells in a given iA3HT resolution can be assigned integer values from 1 to the total number of cells at that resolution using a lexicographical ordering of the character string code form addresses of these cells. This yields a maximally compact integer representation and provides a linear sequence that can be used to traverse

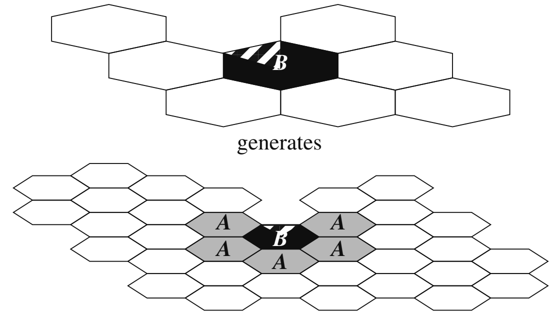

Fig. 22. A3PT closed (type B) generator unfolded on the plane.

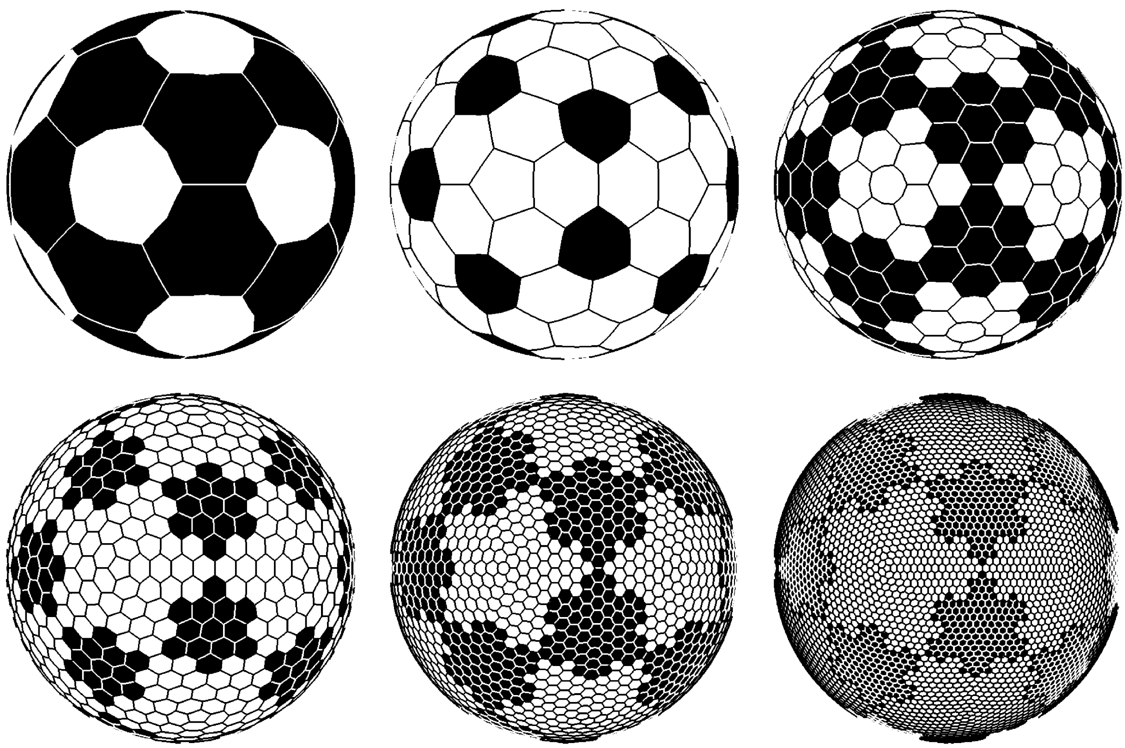

Fig. 23. The first six resolutions of an iA3HT generation pattern on the ISEA3H DGGS. The shading indicates the sub-trees generated by the base tiles.

all of the cells or to map the cells to storage devices that use linear addressing. We call this representation the *sequence indexing code form*.

We note that since each open A3HT or A3PT generator generates only a single center hexagon, every non-zero digit in an iA3HT location code must be followed by a 0 digit. A first order compression of the codes can be achieved by eliminating these redundant 0 digits.

The iA3HT system provides a unique location code for each cell in an icosahedral aperture 3 hexagonal DGGS with many of the desirable characteristics of traditional tree-based path addresses. Since each resolution (beyond the base tiling) is encoded using a single digit, the codes implicitly encode precision metadata. Further, any prefix of an iA3HT location code is guaranteed to be a valid cell at a correspondingly coarser resolution. But because closed iA3HT generators are prefixes for child cells that they do not entirely cover, they cannot be used as a coarse filter for all of their child cells for the proximity operations equality, adjacency, and metric distance. However, we note that all children of a closed iA3HT generator are fully contained in the parent of that generator. Thus in all cases prefixes two resolutions coarser than the target resolution can be used as coarse filters for proximity operations.

6. Quantization algorithms for iMGBT addressing systems

In order for iMGBT-based addressing systems to be useful as location coding systems we must define quantization operators that map between locations on the earth's surface and iMGBT location codes. As previously discussed such algorithms have been developed and implemented for the q2di coordinate system (Sahr, 2002). We can superimpose the unfolded q2di coordinate system given in Fig. 8

on the unfolded MGBT/iA3HT tiling given in Fig. 18. By specifying algorithms that map between q2di coordinates and the corresponding resolution iMGBT/iA3HT location codes we then have a complete quantification algorithm that maps between locations on the earth's surface and iMGBT location codes.

First we discuss the case of mapping a resolution k planar MGBT address to the corresponding resolution k 2di cell coordinates, where both systems share the same origin. We note that, as illustrated in Fig. 24, each successive resolution i MGBT digit corresponds to a resolution i 2di vector V_i. Further, by noting the change in scale in each vector component with each successive resolution we conclude that each resolution i 2di vector V_i corresponds to an appropriately scaled resolution k 2di vector V_{k_i} as follows:

$$V_{k_i} = \begin{cases} V_i \times 3^{(k-i)/2}, & \text{for resolutions} \\ & i \text{ and } k \text{ of same class, and} \\ V_i \times 3^{(k-i-1)/2}, & \text{otherwise} \end{cases}$$

Then a resolution k 2di coordinate C_k corresponding to a resolution k MGBT address is given by:

$$C_k = \sum_{i=0}^{k} V_{k_i}$$

Such an approach works for both full MGBT addresses and for systems that are subsets of the MGBT addressing space (such as the A3HT).

Next we discuss the quantization of a point location on the plane into a resolution k MGBT vector location code. In this case the corresponding resolution k 2di coordinates do not contain sufficient information, since they do not uniquely identify the appropriate resolution $k-1$ parent cell. In this case the point must be quantized into each 2di system from resolution 0 to resolution k. For each resolution i the quantized resolution $i+1$ cell must correspond to one of the seven 2di vectors illustrated in Fig. 24, and a table look-up can therefore be used to determine the corresponding resolution $i+1$ MGBT digit.

Finally, we must discuss the conversion of a resolution k 2di address to a corresponding resolution k address in an addressing system involving a subset of the MGBT addresses, such as the A3HT. One general approach, which can be applied to any aligned incongruent DGGS hierarchy, involves the recursive evaluation of the resolution k footprint of coarser resolution cells. Since aperture 3 hexagon grids are aligned there exists a resolution k cell centered on any given resolution i cell, where $i < k$. Given a resolution i address in an MGBT-subset system, we can specify a resolution k discrete hexagon metric radius R about this central resolution k cell in which all contained cells have as their resolution i prefix the address of this resolution i MGBT cell. Fig. 25 illustrates an example of this for the A3HT system. Given a resolution offset n where:

$$n = \begin{cases} k-i, & \text{for class A/open resolution} \\ & i \text{ A3HT cell, and} \\ k-i+1, & \text{otherwise} \end{cases}$$

The resolution k confirmed footprint radius R of a resolution i A3HT cell is given by:

$$R_i = \begin{cases} 0, & \text{for } n = 0, 1, \text{ and} \\ 1, & n = 2, \text{ and} \\ \frac{(3\times R_{i-1})-1}{2}, & \text{for } n > 2 \text{ and } n \text{ odd, and} \\ 3 \times R_{i-2}, & \text{otherwise} \end{cases}$$

Likewise, we can specify a resolution k radius r about the resolution k cell centered on a resolution i A3HT cell which includes all resolution k cells for which that A3HT cell is a prefix (but which may also include cells that are not children in the subset addressing scheme). Fig. 26 illustrates an example of this for the A3HT system. Given n as above, this radius r of possible child cells is given by:

$$r_i = \begin{cases} 0, & \text{for } n = 0, 1, \text{ and} \\ 1, & n = 2, \text{ and} \\ 2 \times r_{i-1}, & \text{for } n > 2 \text{ and } n \text{ old, and} \\ (3 \times r_{i-2}) + 1, & \text{otherwise} \end{cases}$$

Given a resolution k 2di cell and the origin-centered resolution 0 A3HT cell we can recursively apply this knowledge for each resolution i, $0 < i \leqslant k$, to yield an algorithm for finding the corresponding resolution k A3HT address for the 2di cell. If the resolution k 2di cell is within the con-

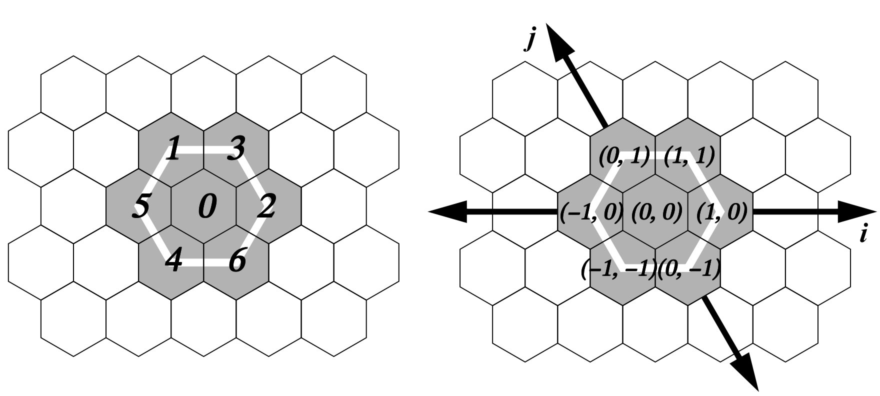

Fig. 24. MGBT digits (on left) correspond to 2di vectors (on right).

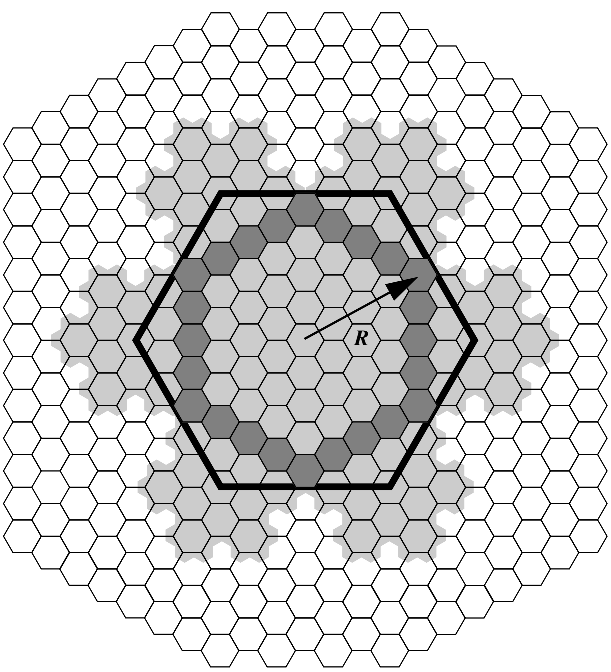

Fig. 25. A resolution i MGBT class B/closed cell (the large black hexagon) generates the indicated gray footprint at resolution $k = i + 4$. The resolution k hexagon radius R about the center point contains only cells contained in the footprint.

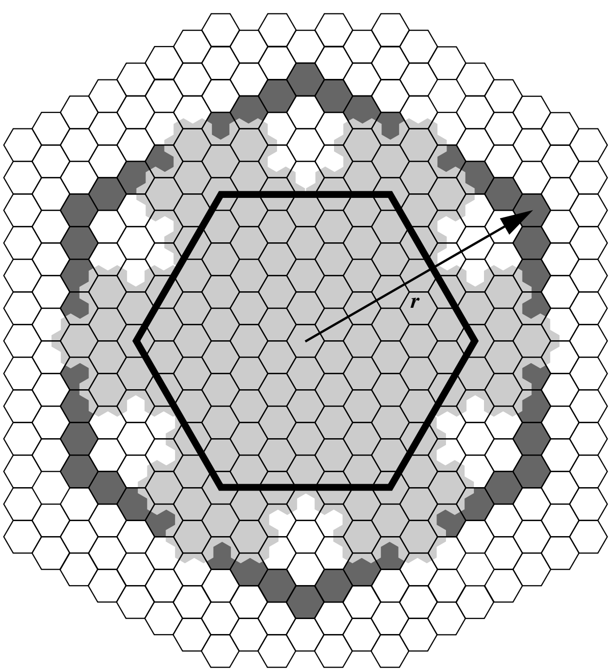

Fig. 26. A resolution i MGBT class B/closed cell (the large black hexagon) generates the indicated gray footprint at resolution $k = i + 4$. The resolution k hexagon radius r about the center point contains the entire footprint.

firmed (radius R_i) footprint of a resolution i child cell, then the address of that child cell is chosen. Otherwise each resolution i child cell that has the 2di cell within its possible (radius r_i) footprint is recursively examined until a confirmed descendent is found. Since MGBT-subset addressing systems tend to be relatively compact and radially symmetric, this approach can yield an efficient algorithm for any such system given appropriate definitions for R_i and r_i.

These algorithms for conversion to/from planar 2di coordinates and planar MGBT and A3HT location coding systems form the basis of algorithms for conversion to/from q2di coordinates and iMGBT/iA3HT addresses. The additional details of these algorithms — which primarily involve book-keeping — are given in (Sahr, 2005).

7. Conclusions

In this paper we have described methods for location-coding icosahedral aperture 3 hexagon DGGSs such as the ISEA3H. The q2di coordinate system provides a pyramid addressing approach that is useful when working with single grid resolutions. This system has an existing base of algorithms and is intuitive enough to make additional algorithm development straight-forward. The iMGBT is a class of path addressing systems that can be used to location code point locations. The iA3HT is a subset of the iMGBT that can be used to location code raster and bucket data. The iMGBT approach represents a true hierarchical location coding system that has the potential to serve as the basis for efficient geospatial algorithms, enabling the use of hexagon DGGSs in a broader range of applications that have traditionally relied on the efficiencies provided by tree-based hierarchies, though more algorithm development will be required before this goal can be completely realized. It is hoped that the location coding systems presented here will facilitate more wide-spread use of this important class of DGGSs by both data structures researchers and end-users.

Acknowledgements

The author would like to thank John Conery, Arthur Farley, A. Ross Kiester, Andrzej Proskurowski, and Denis White for their many helpful comments. Portions of this research were supported by the Canadian Space Agency under contract #9F028-2-4917/03 and by a cooperative research agreement with GRIDS, Limited.

References

Abu-Bakar, S., & Green, R. J. (1996). Detection of edges based on hexagonal pixel formats. In *3rd international conference on signal processing proceedings* (*ICSP-96*)(pp. 1114–1117). Beijing, China.

Burt, P. J. (1980). Tree and pyramid structures for coding hexagonally sampled binary images. *Computer Graphics and Image Processing, 14*, 71–280.

Christaller, W. (1966). Central places in southern Germany. Englewood Cliffs, NJ: Prentice Hall.

Conway, J. H., & Sloane, N. J. A. (1998). Coverings, lattices and quantizers. In *Sphere packings lattices and groups* (pp. 56–62). New York: Springer-Verlag.

Dacey, M. F. (1965). The geometry of central place theory. *Geografiska Annaler, 47*, 111–124.

Dutton, G. (1989). *The fallacy of coordinates. Multiple representations: Scientific report for the specialist meeting*. Santa Barbara, CA: National Center for Geographic Information and Analysis.

Fredkin, E. (1960). Trie memory. *Communications of the ACM, 3*(9), 490–499.

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

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

Gibson, L., & Lucas, D. (1982). Spatial data processing using generalized balanced ternary. In *Proceedings of the IEEE computer society conference on pattern recognition and image processing*(pp. 566–571). Las Vegas, NV: IEEE Computer Society.

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

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

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

Kimerling, A. J., Sahr, K., White, D., & Song, L. (1999). Comparing geometrical properties of global grids. *Cartography and Geographic Information Science, 26*(4), 271–287.

Knuth, D. (1998). *The art of computer programming. Seminumerical algorithms* (2). Menlo Park, CA: Addision-Wesley.

Luczak, E., & Rosenfeld, A. (1976). Distance on a hexagonal grid. *IEEE Transactions on Computers, C-25*(5), 532–533.

Rothman, D. H., & Zaleski, S. (1997). *Lattice-gas cellular automata: Simple models of complex hydrodynamics*. Cambridge: Cambridge University Press.

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

Saff, E. B., & Kuijlaars, A. (1997). Distributing many points on a sphere. *Mathematical Intelligencer, 19*(1), 5–11.

Sahr, K. (2002). DGGRID: User documentation for discrete global grid software. Retrieved November 1, 2007, from http://www.sou.edu/cs/sahr/dgg/dggrid/docs/dggriddoc31.pdf.

Sahr, K., White, D., & Kimerling, A. J. (2003a). Geodesic discrete global grid systems. *Cartography and Geographic Information Science, 30*(2), 121–134.

Sahr, K., Peterson, P., & Lutterodt, L. (2003b). The aperture 3 hexagon tree. Unpublished technical report. GRIDS, Limited, Kingston, Ontario, Canada.

Sahr, K. (2005). Discrete global grid systems: A new class of geospatial data structures. Unpublished Ph.D. thesis, University of Oregon, Eugene, OR.

Samet, H. (1989). *The design and analysis of spatial data structures*. Menlo Park, CA: Addison-Wesley.

Samet, H. (1990). *Applications of spatial data structures: Computer graphics, image processing and GIS*. Menlo Park, CA: Addison-Wesley.

Snyder, W., Qi, H., & Sander, W. (1999). A coordinate system for hexagonal pixels. *Proceedings of SPIE, the international society for optical engineering, 3661*, 716–727.

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

van Roessel, J. (1988). Conversion of cartesian coordinates from and to generalized balanced ternary addresses. *Photogrammetric Engineering and Remote Sensing, 54*(11), 1565–1570.

Verbrugge, C. (1997). Hex grids. Unpublished manuscript, McGill University, Montreal, Quebec, Canada.

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

Wuthrich, C. A., & Stucki, P. (1991). An algorithmic comparison between square- and hexagonal-based grids. *CVGIP: Graphical Models and Image Processing, 53*(4), 324–339.