A Survey of Digital Earth Representation and Visualization

Mahdavi-Amiri, Ali; Alderson, Troy; Samavati, Faramarz

http://hdl.handle.net/1880/50407
technical report

Downloaded from PRISM: https://prism.ucalgary.ca
A Survey of Digital Earth Representation and Visualization

Ali Mahdavi-Amiri†, Troy Alderson‡, Faramarz Samavati§

†University of Calgary

Abstract
The creation of a digital representation of the Earth and its associated data is a complex and difficult task. The incredible size of geospatial data and differences between data sets pose challenges related to big data, data creation, and data integration. Advances in globe representation and visualization have made use of Discrete Global Grid Systems (DGGSs) that discretize the globe into a set of cells to which data are assigned. DGGSs are well studied and important in GIS, OGC, and Digital Earth communities. However, DGGSs have not been introduced very well to computer graphics community. In addition, there are many advanced techniques related to geospatial data creation and representation that might be very useful to Digital Earth community. In this paper, we provide an overview of DGGSs and their use in digitally representing the Earth as well as the list of current Digital Earths and their method of Earth representation. In addition, we present key research areas and related papers in computer graphics that are useful for a Digital Earth framework. Moreover, we list a number of applications of Digital Earths and their related works.

Categories and Subject Descriptors (according to ACM CCS): H.2.8 [Information Systems]: Database Applications—Spatial databases and GIS

1. Introduction

The Earth is immense, and data about the Earth is similarly immense. The field of GIS exists to gather, store, integrate, process, and distribute this data, traditionally performed by individual GIS experts. Several petabytes of geospatial data occupy GIS servers around the world, and more continues to be generated every day.

Geospatial data is typically found in one of three forms: raster data (e.g. satellite imagery), feature data (e.g. road networks and nation boundaries, usually represented as vector data), and 3D geometry. The resolution and size of these data are usually quite high, depending on the data capture technologies used, and are constantly growing as such technologies improve.

One of the main challenges faced by GIS experts is the task of data integration. These geospatial data are gathered and processed by disparate organizations and stored in various file formats at different resolutions. The traditional model of GIS, in which GIS experts clean, process, and integrate the data, is unsustainable in the face of the ever increasing flow of data.

Another challenge lies in handling the spherical nature of the Earth. The surface of the Earth forms a non-Euclidean spherical space, and any processing that is performed on the Earth’s surface data must take this into account.

A technique that has emerged to facilitate solutions to these challenges is the Digital Earth framework (aka: Digital Globe, and Virtual Globe), which uses the curved Earth itself as a reference model for geospatial data. As can be inferred from the name, the Digital Earth framework represents the Earth within a digital environment. What makes such a representation possible are Discrete Global Grid Systems (DGGSs), discretizations of the Earth into indexed (mostly regular) cells, each of which represents a region of the Earth and to which associated data is assigned.

While latitude/longitude parametrization is the traditional method of choice for discretizing the globe [SWK], the areas and 3D geometry of the cells vary widely over the latitudes, which complicates data processing, and singularities exist at the poles, causing neighboring cells to degenerate into triangles. More advanced DGGSs aim for uniform, equal-area...
cells. Typically, a spherical polyhedron is used to approximate the Earth, which is then refined by some refinement method and projected onto the spherical Earth.

DGGSs may be characterized according to the polyhedron used to approximate the Earth, the refinements applied to this polyhedron, the shapes of the resulting faces, the projection method that maps the faces to cells on the spherical Earth, and the indexing method used to refer to these cells. We present in this paper a survey of DGGSs as characterized by these factors, in addition to a brief overview of state of the art DGGSs currently in use and applications of globe representation and visualization.

2. Discrete Global Grid Systems

A Discrete Global Grid System, as defined in [SWK], is a series of Discrete Global Grids of increasingly fine resolution, where each Discrete Global Grid is a partition of the surface of the Earth into cell regions.

[Goodo0] discusses what a Digital Earth framework is and lays out a number of requirements for Digital Earth applications and how DGGSs can contribute to these applications. In particular, the consistency, hierarchical structure, unique addresses (indices) for geographical regions, and explicit representation of resolution in DGGSs facilitate solutions to the many challenges faced by Digital Earth and simplify a number of implementation issues.

![Figure 1](image.png)

Figure 1: (a) The Earth discretized into a latitude/longitude grid. A line of longitude (b) and latitude (c) are highlighted on the Earth. (d) Cells close to the pole are triangular.

2.1. Cell Creation

The quality of the DGGS cells (i.e. their angular and areal distortion and regularity) naturally depends on the method(s) used to create those cells. While the cells can be calculated directly on the surface of the sphere, as with lat/long parametrization, the quality tends to be much improved when the cells result from a process of refining and projecting the faces of an initial polyhedron that approximates the Earth. Sahr [SWK] refers to such DGGSs as Geodesic DGGSs.

2.1.1. Initial Polyhedron

A number of different polyhedrons have been used for initial approximations of the Earth. The platonic solids — the tetrahedron, cube, octahedron, dodecahedron, and icosahedron — in addition to the truncated icosahedron are among the most common choices.

The octahedron, for instance, can be projected to the sphere in such a way that each of its faces corresponds to an octant of the latitude/longitude spherical coordinate system, as in [Whi00, H.d98, GS92, Vin06, BTC10], whereas the simplicity of the tetrahedron has motivated its use in [CR11]. The cube benefits from alignment with the Cartesian coordinate system and the prevalence of compatible data structures, such as quadtrees [AS00, GRS12, GHB’05].

While the dodecahedron features lower angular distortion under equal area projection (such as Snyder projection [Sny92]) compared to the other platonic solids [WEE74], pentagon-to-pentagon refinement is undefined, hence the dodecahedron is not a popular choice for approximating the Earth. A potential solution is to triangulate each polygon by splitting the dodecahedron’s faces into five isosceles triangles.

By comparison, the truncated icosahedron is very commonly used to approximate the Earth [FT90, Sah08, Whi00, Pet06, VZ09]. The icosahedron, which introduces lower distortion than the tetrahedron, cube, and octahedron, can be refined into the truncated icosahedron, which features a low amount of areal distortion under equal area projection [Sny92, WK09].

The faces of these polyhedrons generally cover large areas of the Earth’s surface. In order to increase the resolution of the DGGS, polyhedral refinements may be employed.

2.1.2. Polyhedral Refinement

Polyhedral refinement methods produce a set of fine faces from a set of coarse faces, and can be used to construct more cells in a DGGS by introducing more faces into the approximating polyhedron. Refinement methods are categorized according to their aperture factor: the number of fine faces produced per coarse face. The fine cells produced by a refinement may be assigned to coarse cells as children, defining a hierarchy on the cells and a hierarchical traversal between parent and child cells. In addition to the aperture factor of a refinement, other characteristics may be defined, such as congruency and alignment.

A refinement is called congruent when a coarse cell can be formed by a union of finer cells (see Figure 2), and a DGGS is called congruent when its employed refinement is congruent. For example, quadrilateral 1-to-4 refinement (as shown in Figure 2 (a)) is congruent while hexagonal 1-to-3 refinement is incongruent (see Figure 2 (b)). Assigning a set of fine cells to be the children of a coarse cell is trivial in congruent refinements, since the children are fully covered by their parents and hierarchical traversal queries are simplified. Incongruent refinements, by contrast, require more thought to be put into which child belongs to which parent,
as each fine cell can have several potential parents (Figure 2 (c)).

Alignment refers to the case in which a coarse cell shares a centroid with a fine cell. A refinement for which this is the case is called aligned, as is any DGGS that employs it, otherwise they are unaligned. Alignment in a DGGS implies that traversing from one resolution to another always results in an improvement in accuracy, and so is quite a beneficial property to have. Accuracy, here, is a measure of how well a point may be represented by a cell, and is found as the distance between \( p \) and the centroid of its containing cell at resolution \( r \), which we will denote as \( m_r \). As the resolution \( r \) increases under an aligned refinement, this distance decreases, i.e. \( d_r \leq d_{r+1} \) where \( d_r = ||p - m_r||_2 \). Figure 3 illustrates a comparison between unaligned 1-to-4 refinement and aligned 1-to-2 refinement.

In addition to these properties, a low aperture factor is considered desirable for a DGGS refinement, since such a refinement increases the number of faces at a low rate through the resolutions. Hence, more resolution levels are produced under a fixed maximum number of faces and, therefore, a smoother transition between resolutions may be achieved. However, it may not be possible to encapsulate all of these properties (low factor, alignment, and congruency) in a single refinement method; as a result, one or more property is usually sacrificed.

For example, it is possible to define an aligned 1-to-2 refinement for quadrilateral cells which has the slowest rate of growth (Figure 4 (a)) [MABS13], but which is not congruent, while the commonly employed 1-to-4 refinement for quadrilateral cells (Figure 4 (b)) is not aligned [AS00, GHS*05]. Quadrilateral 1-to-9 refinement is both aligned and congruent but has a larger factor (Figure 4 (c)) [GRS12]. Triangular cells in DGGSs tend to be refined by an aligned and congruent 1-to-4 refinement (see Figure 4 (d)) [H.d98, GS92, GHS*07].

Figure 3: Accuracy is measured in terms of the distance \( d_r \) from a point \( p \) to the centroid \( m_r \) of its containing cell through the resolutions. (a) \( d_0 \) is the distance between \( m_0 \) and \( p \), illustrated as a red square. (b) Under aligned refinements, the accuracy is at least as good as that of the previous resolution. (c) Under unaligned refinements, the accuracy may be decreased \( (d_1 > d_0) \).

Aside from quadrilateral and triangular cell refinements, there exist a number of types of hexagonal refinement [MAHS14], each of which exhibits characteristics that can prove useful in DGGS applications and in which growing interest has developed [Sah12, MS05]. For example, Sah [Sah08] uses hexagonal 1-to-3 refinement, which has the lowest factor possible for hexagonal refinements, on the icosahedron while Vince [Vin06] uses the same refinement on the octahedron. While this and other hexagonal refinements introduce a rotation in the lattices of two successive resolutions [IDS04], hexagonal 1-to-4 refinement produces rotation free lattices at all levels of resolution, simplifying hierarchical analysis [SAM68, Thu97]. Two types of 1-to-4 hexagonal refinement can be combined to obtain a better hierarchy, as performed by Tong et al. [TBW10], but, like the aforementioned hexagonal refinements, it is incongruent. 1-to-7 refinement comes close to congruency, covering the coarse hexagons with fine hexagons better than others.

Once the initial polyhedron has been refined and the desired number of faces have been achieved, the faces must be projected onto the surface of the Earth to form DGGS cells.

2.1.3. Projection

Spherical projections have a rich history in the field of cartography, and have been used to represent the spherical Earth on flat maps. Types include conformal, gnomonic, or equal
area [GK06]. Inverting such projections allow flat shapes, such as polyhedral faces, to be mapped to spherical DGGS cells. However, any mapping from spherical space to Euclidean space results in undesirable distortions. One may try to reduce the distortions in a spherical projection, and many different projections exist, each with different levels of distortion.

Traditional cartographic projections are transformations from a point on the Earth to a point on a 2D map. A projection can be represented as a function:

\[
p_m = F(p_s)
\]

where \( p_s \) lies on the sphere, and \( p_m \) is located on the 2D map (see Figure 1.3). The forward projection \( F \) can be inverted into \( F^{-1} \), which defines a mapping from a point on the 2D map to a point on the sphere.

The obvious method for such a projection is lat/long, or spherical coordinate, conversion. This method cuts the Earth (along a meridian) and unfolds it onto a rectangular map with the lines of latitude (given by fixing angle \( \phi \)) and longitude (given by fixing angle \( \theta \)) serving as the two main axes of the 2D domain (see Figure 6). This 2D domain and the sphere are related through Equations 1 and 2, where \( R = \sqrt{x^2 + y^2 + z^2} \) is the radius of the sphere (see Figure 7).

\[
F(x,y,z) = \begin{pmatrix}
\theta = \tan^{-1}(y/x) \\
\phi = \cos^{-1}(z/R)
\end{pmatrix}
\]

(1)

\[
F^{-1}(\theta,\phi) = \begin{pmatrix}
R\cos(\theta)\sin(\phi) \\
R\sin(\theta)\sin(\phi) \\
R\cos(\phi)
\end{pmatrix}
\]

(2)

As demonstrated in Figure 7, perfect squares on the 2D domain are mapped to distorted quadrilateral and even triangular cells on the sphere. In general, when given a function \( G(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} \) that maps \( (u,v) \in \Omega \subset \mathbb{R}^2 \) to a continuous surface \( S \subset \mathbb{R}^3 \), we can study the distortion behavior of \( G \) using its first fundamental form, whose singular values \( \sigma_1 \) and \( \sigma_2 \) can be used to evaluate the distortion of \( G \). Four possibilities exist (\( \nu \) is a constant) [HLS07, SM01]:

- \( \sigma_1 = \sigma_2 = \nu \) implies \( G \) is isometric or distance-preserving,
- \( \sigma_1 = \sigma_2 \) implies \( G \) is conformal or angle-preserving,
- \( \sigma_1\sigma_2 = \nu \) implies \( G \) is equivalent or angle-preserving,
- and \( \sigma_1 = \nu \) implies \( G \) is stretch-preserving for all \( (u,v) \in \Omega \).

Unfortunately, it is impossible to define an isometric mapping between the sphere and 2D maps [AT12], although equal area, conformal, and stretch preserving mappings exist. As noted by White et al. [WKO92], equal area projections are the best fit with DGGSs, as these systems use planar or piece-wise planar (polyhedral) domains to sample the spherical surface of the Earth. Examples of equal area projections include Lambert’s cylindrical and azimuthal, Mollweide’s, and Werner’s equal area projections (see Figure 8) [Sny97, Har12].

\[
\begin{array}{ccc}
(a) & (b) & (c)
\end{array}
\]

Figure 8: (a) Werner, (b) Mollweide, and (c) Lambert (cylindrical) projections.

Among these projections, Lambert azimuthal equal
area projection is particularly interesting since it forms the base of some equal area projections for polyhedrons. Given a point \( c \) on the sphere \( S \) and the plane \( \rho \) that is tangent to \( S \) at \( c \), a point \( p_c \) on the sphere is projected to a point \( p_{\rho} \) on the 2D map by taking \( p_{\rho} \) to be the intersection of \( \rho \) with a circle with center \( c \) that passes through \( p_c \) and is normal to \( \rho \) (see Figure 9). Note that \( c \) is projected to itself on \( \rho \) and its antipode is excluded from the projection as the intersecting circle is not unique.

Polyhedral globe projections, which are functions \( F \) that map a point \( p_s \) on the sphere to a point \( p_{\rho} \) on some polyhedron, may also be defined. The inverses of these functions map \( p_{\rho} \) from the polyhedron back to \( p_s \) on the sphere.

**Snyder’s equal area projection**, for instance, is defined for all platonics solids and the truncated icosahedron [Sny92]. This aspect of Snyder projection is rather unique, as most polyhedral projections are specifically defined for only one polyhedron. In this projection, function \( F \) maps point \( P \) on the sphere to a polyhedron. To find such a function, Snyder employs Lambert azimuthal equal area projection individually for each face. To perform the projection, the problem is initially reduced for a right triangle considering the symmetry property of polyhedrons (see Figure 11). Then a scaling factor is found between the radius of the sphere (spherical polyhedron) and the sphere that circumscribes the polyhedron. Finally, a triangle on the polyhedron matching its corresponding spherical triangle encompassing point \( P \) is generated. Final equation of Snyder’s projection is presented in a closed form provided in Equation 3 in which ratio \( \rho \) and angle \( \tilde{\alpha} \) are defined by a set of trigonometric functions and known constants that are provided in [Sny92].

\[
\begin{align*}
x &= \rho \sin \tilde{\alpha} \\
y &= \rho \cos \tilde{\alpha}
\end{align*}
\]  

(3)

While Snyder projection has a simple closed form, its inverse calculation requires finding the roots of a nonlinear equation [Sny92]. Snyder suggests the use of the Newton-Raphson iterative technique to determine the inverse projection, although this process can be slow. To reduce inefficiency, Harrison et al. [HMAS11, HMAS12] optimized the inverse of Snyder projection by providing initial estimates for the Newton-Raphson technique that are close to the roots of the nonlinear equation, found using polynomial curve fitting.

**Roška and Plonka’s Equal Area Projection** [RP11] defines a mapping from the cube \( \Omega \) to the unit sphere \( S \), both of which are centered at the origin and have equal surface area (i.e. each edge of the cube has length \( a = \sqrt{2}/3 \)). The sphere \( S \) is divided into six equal partitions, each corresponding to a face of the cube \( \Omega \). The projection proceeds by mapping each face of \( \Omega \) to a curved square on a plane tangent to \( S \) using an equal area bijection (say, \( T \)), and then projecting the curved square to a partition of \( S \) using inverse Lambert azimuthal equal area projection (say, \( F^{-1} \)). See Figure 11 for an illustration. This approach was later generalized to the octahedron in [RP12].

Additional polyhedral globe projections include the slice-and-dice approach [vLS06], in which the projection consists of an equal area partitioning (“slice”) step and an equal area positioning (“dice”) step; the equal area projection for cubes described in [AS00, Alb06], in which the sphere is divided into size equal area segments that are each projected to a face of the cube using Lambert cylindrical equal area projection; and the equal area projection for octahedrons from [HR14], which slices the sphere into eight sections and establishes a geometric relationship between each section of the sphere and each face of the octahedron.

Note that projections only affect the geometry of the refined polyhedron, and so are data structure independent. The shapes of the cells resulting from the cell creation process, and the data structure(s) used to store them, hence depend only on the initial polyhedron and refinement method.
Several different cell shapes — triangles, quads, or hexagons — can be used to discretize the Earth, each with its own advantages and disadvantages.

As they work well together with Cartesian coordinates, quadrilateral cells are fairly easy to use. DGGSs and spherical representations that have employed them include the works of [CGG03, Gre86, CGG07, Gri02, CR11, Alb06]. However, an important issue with quadrilateral cells is that they are unable to cover the surface of the Earth without considerable distortion.

Triangular cells are simple to use, efficient to render, and are compatible with the faces of polyhedrons such as the icosahedron, dodecahedron, or tetrahedron. For example, triangular cells are used by Dutton [H.d98] with the octahedron and Lee and Samet with the icosahedron [LS98] to represent the Earth.

As discussed by Sahr [Sah11], hexagonal cells are preferable in some applications due to uniform adjacency [HJ05], regularity, and efficiency in sampling [KPKPS89], despite greater challenges in refinement and indexing. As a result, hexagons are widely employed in representations of the Earth [Sah08, PYX15, Vin06, VZ09, TBW*13, Sah12].

Regardless of the cell shape that results, a mechanism for efficiently accessing the data stored in each cell is a requirement for a functional DGGS, and for this purpose several different methods have been proposed.

2.3. Cell Indexing

While hierarchical data structures such as quadtrees are the method of choice for establishing efficient access to hierarchical data in other problem domains, and have been applied within the Digital Earth setting [AS00, FT90], the sheer immensity of the Digital Earth causes tree structures that record node dependencies to become prohibitively expensive. Hence, the role of providing this access is often served by cell indexing methods. For each cell that exists in a DGGS, an indexing method assigns to it an index that uniquely identifies that cell. Using this index, a hierarchical traversal may be performed and the cell’s location (typically its centroid) and its resolution may be determined. Furthermore, index i may serve also as a reference into a data structure or database in order to retrieve data associated with the cell.

Cell indices can take many forms, be they 1D strings of letters and digits or nD coordinate values (with n coordinate axes defined on the faces of the polyhedron). Although various types exist, as discussed in [AMA14] most indexing methods for DGGSs are derived using three general indexing mechanisms: hierarchy-based, space-filling curve (SFC) based, and coordinate-based.

**Hierarchy-based indexing** relies on the hierarchy of cells generated by the application of refinements on the initial polyhedron. Under these indexing methods, the fine child f of a coarse cell c inherits the index of c as a prefix for its own index. Formally, if cell c at resolution r has index \( Id_0d_1d_2\ldots d_{r-1} \), its children \( f_i \) receive indices \( Id_0d_1d_2\ldots d_{i-1}i \). Note that if the maximum number of children per cell is \( b \), then the digits \( d_i \in [0, b - 1] \) and every cell’s index is a base \( b \) integer. Integer \( b \) is hence considered the base of the indexing method, and may be used to define algebraic operations on indices, such as conversion to and from the Cartesian coordinate system, or neighborhood finding [VZ09, SWK, GS92].

![Figure 12: (a) Hierarchy-based indexing for 1-to-4 refinement. (b) Hierarchy-based indexing method for aligned quadrilateral 1-to-9 refinement.](image)

In order to use hierarchy-based indexing methods within a DGGS, the faces of the initial polyhedron must first be indexed (arbitrarily, using letters or digits), after which the faces of the refined polyhedron may be indexed using the hierarchical indexing method. For instance, the six faces of the cube can be indexed using six different letters and each descendant face inherits the index of its parent (see Figure 13). Works on DGGSs that have made use of hierarchy-based indexing include [GHBR05] (quadrilateral cells under 1-to-4 refinement on an initial cube), [H.d98, SGF07, GS92] (triangular cells under 1-to-4 refinement), and [TBW*13] (hexagonal cells under 1-to-4 refinement on an initial icosahedron).

![Figure 13: Hierarchy-based indexing given an initial polyhedron. (a) The faces of the initial polyhedron are indexed using letters. (b) The faces that result from refinement inherit the parent index. (c) Faces are projected to the sphere.](image)

Another means of indexing cells is to run a curve through all the cells and index them according to the order in which they are intersected. Such curves may be denoted by \( f(t) : T \subset R \rightarrow Q \subset R^2, t \in T \).

Space Filling Curves (SFCs) are appropriate choices to index cells at increasingly fine resolutions. SFCs are...
recursively-created curves whose range covers an entire space, typically providing a surjective and continuous mapping from \( T \subset R \) to \( Q \subset R^2 \). Using a 2D SFC that visits all the cells of \( Q \) after refinement, we can define an indexing on the cells of \( Q \) by discretizing \( T \). The 1D index \( i \) for each cell in \( Q \) is defined on domain \( T \) (by taking a unit step on \( T \), \( i \) is incremented) and \( i \)’s corresponding cell in \( Q \) is returned by the mapping \( f(i) \in Q \).

Commonly encountered SFCs include the Hilbert, Peano, Sierpinski, and Morton (Z) curves, which are illustrated in Figure 14. These curves typically begin as a simple initial geometry defined on a simple domain, which is then refined and the simple geometry repetitively transformed to cover the entire refined domain. A set of production rules govern the repetition of the simple geometry through the refinements. The production rules of the Hilbert curve, for example, are provided in Equation 4 and illustrated in Figure 15. The Hilbert curve’s geometry starts on a simple two by two domain, which is then refined by an unaligned 1-to-4 refinement. Typically, if the initial geometry covers \( i \) cells, a 1-to-\( i \) refinement is suitable for application on the domain. Hence, each SFC may be associated with a refinement.

\[
\begin{align*}
H & \leftarrow A \uparrow H \rightarrow H \downarrow B \\
A & \leftarrow H \rightarrow A \uparrow A \leftarrow C \\
B & \leftarrow C \leftarrow B \downarrow B \rightarrow H \\
C & \leftarrow B \downarrow C \leftarrow C \uparrow A
\end{align*}
\]

(4)

The resolution of each cell can be determined directly from the index, as with hierarchy-based indexing methods, if the indices are taken to be base \( b \) integers. The base \( b \) of the indexing method is usually taken to be \( i \) or \( \sqrt{i} \) if 1-to-\( i \) refinement is associated with the SFC curve (see Figure 16). For instance, the refinement associated with the Hilbert and Morton curves is 1-to-4; therefore, indices in base four or two are appropriate for Hilbert and Morton curves. Then, given a cell at resolution \( r \), the cell’s index in base \( i \) will have a length of \( r \) and in base \( \sqrt{i} \) will have a length of \( 2r \). Unlike hierarchy-based indexing, redundant bits are unnecessary in SFC-based cell indices, as every cell resulting from these refinements can be indexed between 0...0 and \( (b-1) \ldots (b-1) \). Figure 16 illustrates such an indexing for the Hilbert and Morton curves using bases two and four.

Works that have made use of SFC-based indexing for DGGSs and terrain rendering include [JB05, Whi00], which used Morton indexing on cells resulting from 1-to-4 refinements on the icosahedron and octahedron (see Figure 17 for an illustration of [Whi00]’s method on an unfolded octahedron); and [BIG00], which used the Sierpinski SFC to index triangular cells refined with a factor of two.

\[
\begin{align*}
\begin{array}{cccc}
00 & 01 & 02 & 03 \\
04 & 05 & 06 & 07 \\
08 & 09 & 0A & 0B \\
0C & 0D & 0E & 0F
\end{array}
\end{align*}
\]

(5)

Figure 16: (a) Hilbert SFC-based indexing in base 2 and base 4. (b) Morton SFC-based indexing in base 2 and base 4.

\[
\begin{align*}
\begin{array}{cccc}
00 & 01 & 02 & 03 \\
04 & 05 & 06 & 07 \\
08 & 09 & 0A & 0B \\
0C & 0D & 0E & 0F
\end{array}
\end{align*}
\]

(6)

Figure 17: Morton indexing on the octahedron. (a) Unfolded octahedron. (b) Two triangles are merged to obtain a diamond, which is then recursively refined. (c), (d) Morton SFC-based indexing on the diamond.

A third method of indexing the cells of a DGGS is to define an \( m \)-dimensional coordinate system (typically 2- or 3D) using a set of axes \( U_1, \ldots, U_m \) that spans the entire space on which the cells lie, known as \textit{coordinate-based indexing}. A subscript \( r \) may be appended to the index in order to record the resolution of a cell. Each index, then, is of the form \((i_1, i_2, \ldots, i_m)_r\), where the \( i_j \) are integer numbers indicating the number of unit steps taken along the axis \( U_j \) and \( r \) is the resolution.

A simple example of such an indexing uses the axes of the Cartesian coordinate system in order to index a quadrilateral domain, as illustrated in Figure 18 (a). Another, seen in [Vin06, BTC10], is a 3D coordinate indexing system in which the Barycentric coordinate of each cell is taken to be...
its index. In [MAS12], a 2D indexing method is applied after unfolding the initial polyhedron onto a 2D domain.

In order to simplify the indexing method, it is possible to define local coordinate systems for each face of the initial polyhedron rather than a global coordinate system over all the faces [MAS14, MAHS14]. In this case, indices are augmented with additional information in order to specify which of these initial faces a coordinate value is defined on. Hence, a cell \((i_1, i_2, \ldots, i_m)\) in the coordinate system of face \(f\) will be given index \([f, (i_1, i_2, \ldots, i_m)]\).

**Figure 18:** (a) Coordinate-based integer indexing based on Cartesian coordinates. (b) A cube. (c) The cube unfolded with coordinate systems assigned to each face. (d) The indices of some cells are shown after one step of 1-to-4 refinement.

### 3. State-of-the-art DGGSs

The variety of implementation possibilities for DGGSs, and the significant benefits they offer to geospatial applications, have resulted in a number of proposed and realized DGGSs in the literature and industry. We briefly describe a selection of these systems in this section, including some DGGSs that construct cells directly on the sphere or through an intermediate domain.

One example of a DGGS is the HEALPix software package [GHB+05], which stands for Hierarchical Equal Area isoLatitude PXelation of a sphere. In the HEALPix system, the Earth is approximated using 12 initial curvilinear quadrilaterals that are then refined using a congruent 1-to-4 refinement (see Figure 19). The faces are then projected to the sphere via a combination of Lambert cylindrical equal area projection (for equatorial regions) and Collignon equal area projection (for polar regions). A base 2 hierarchy-based indexing method is used to index the cells, although a coordinate-based method based on integer Cartesian coordinates has also been proposed.

A very similar approach to that of Heal-Pix is used by the SCENZ-Grid system, but rather than a congruent 1-to-4 refinement, a congruent and aligned 1-to-9 refinement is applied on the cube to establish the multiresolution representation [GRS12]. Its indexing method is also hierarchy-based, but in base 9 (see Figure 20).

Dutton and Goodchild, two pioneers in designing global (Digital Earth) models, have proposed their own DGGSs. Dutton’s Quaternary Triangular Mesh (QTM) [H.d98] creates cells using congruent and aligned 1-to-4 quaternary refinement on the faces of the octahedron (see Figure 21). The initial faces of the octahedron are indexed 1 through 8, with their children (0 through 3) indexed using a base 4 hierarchy-based scheme. A specific projection called Zenithial Orthotriangular Projection (ZOT) is designed for QTM which is neither equal-area nor conformal [Dut91].

**Figure 19:** The HEALPix system. (a) Repeated refinement and projection on the sphere. (b), (c) Quadrilateral cell indexing system. The index of a coarse quad is inherited by its children.

**Figure 20:** The SCENZ-Grid system. (a) Repeated refinement and projection on the sphere. (b), (c) Quadrilateral cell indexing system.

**Figure 21:** The QTM system. (a) An octahedron embedded in a sphere. (b), (c) The indexing method applied to the unfolded faces of the octahedron.

Goodchild’s Hierarchical Spatial Data Structure (HSDS) system also estimates the globe using the octahedron, though refined under a congruent and aligned 1-to-4 triangular refinement applied directly on the sphere (area is not preserved) after associating each face of the octahedron with a spherical triangle. The indexing is similar to that of QTM, but uses a continuous ordering of the children of each cell (see Figure 22). The employed projection in HSDS is also neither equal-area nor conformal.

A number of DGGSs with hexagonal cells have arisen due to the beneficial properties of hexagonal cells in Earth
representation, but due to the incongruent nature of hexagonal refinements, indexing these cells is not straightforward.

**PYXIS indexing** is a hierarchy-based method designed for incongruent 1-to-3 aligned refinement on an icosahedron. The initial application of this refinement results in a truncated icosahedron, the faces of which each receive an alphanumeric index and which are projected to the sphere via Snyder’s equal area projection [VZ09, Sny92] (see Figure 23).

Figure 23: Hexagonal cells at three successive resolutions in PYXIS’s system.

Figure 24: (a) Type A cells (orange) surround a type B cell (black) with index b. (b) The children of the cells illustrated in (a). (c), (d) The descendants of type A and B cells, respectively, after five successive refinements. Notice the fractal boundary developing at the finer resolutions.

The descendants of these faces are categorized into two types, A cells and B cells, such that each type B cell is surrounded by six type A cells. At each level of refinement, each type A cell is assigned one child—the type B cell with which it shares a centroid—and each type B cell is assigned to be the parent of the seven cells it partially or wholly covers. The cell with which it shares a centroid (which is wholly covered) is taken to be type B while the six cells surrounding it (that are partially covered) are taken to be type A. As shown in Figure 24, each cell’s descendants at a fine resolution form fractal shape boundaries that fit together perfectly to cover the entire spherical icosahedron.

A similar hierarchy-based approach was proposed by Sahr [Sah08], who also suggests a coordinate-based pyramid indexing based on hexagonal coordinate systems. The pyramidal scheme, however, was only developed for a single resolution, and cannot index descendant cells resulting from an arbitrary refinement.

Sahr later proposed **Central Place Indexing (CPI)**, a hierarchy-based indexing scheme designed for hexagonal cells that are refined by aligned and incongruent 1-to-7, 1-to-4, or 1-to-3 refinements [Sah12]. Under this indexing, using a 1-to-i refinement implies a scaling factor of $\sqrt{i}$ should be employed to create finer cells, and it is possible to combine different refinements to generate resolutions. Naturally, the order of the refinements matters, as the scaling factor for each refinement is different.

**Octahedral Aperture 3 Hexagonal Discrete Global Grids** (OA3HDGG) employ 1-to-3 hexagonal refinement on the octahedron, for which a coordinate-based indexing based on barycentric coordinates is used to index the cells [Viu06]. This indexing assigns to the vertices the coordinates $(±1,0,0)$, $(0,±1,0)$, and $(0,0,±1)$. Throughout the resolutions, the barycenter of each cell with respect to these coordinates is taken to be its index. This method can be modified to index the cells of **Octahedral Aperture 4 Hexagonal Discrete Global Grids** (OA4HDGG), which employ 1-to-4 hexagonal refinement [BTC10]. Snyder’s equal area projection [Sny92] can be used in both cases to obtain equal area cells on the sphere.

Hexagonal cells see further use in the **Hexagonal Quaternary Balanced Structure** (HQBS), another type of DGGS that refines an icosahedron under 1-to-4 refinement and projects cells via Snyder’s equal area projection [TBW13, 20113]. HQBS defines a triangular hierarchy for hexagonal cells aligned with the edges of the icosahedron. In order to establish such a hierarchy, two variations (aligned and unaligned) of hexagonal 1-to-4 refinement are combined to index lattice points that result from refinements, for which a base 4 hierarchy-based mechanism is used. Each cell receives the index of the point that is located at its centroid (see Figure 25).

Figure 25: Combining two types of 1-to-4 refinement leads to a hierarchy-based indexing method for hexagons.

Other methods besides the polyhedron-refinement-projection paradigm may be used to define a DGGS. The
Spheroid Degenerated-Octree Grid (SDOG) is a system of data representation that uses a volumetric discretization of the sphere. This representation is primarily designed to represent the global lithosphere (crust and a portion of upper mantle of the Earth) [YWZG12]. In this system, the sphere is initially divided into eight octants, each associated with a degenerate octree for further levels of subdivision. (These octrees are degenerate due to the existence of the triangles near to the poles; see Figure 26).

Indexing of the grids is performed using two methods: Single Hierarchical Degenerated Z-curve Filling (SDZ), and Multiple Hierarchical Degenerated Z-curve Filling [JqLx09]. Both methods are based on a modified Z-curve defined on a congruent, unaligned 1-to-4 refinement. The SDZ scheme indexes a single resolution of the octree in base 10 (Figure 27), whereas the MDZ scheme provides a base 8 hierarchical indexing (Figure 28).

A latitude/longitude tiling of the Earth (resulting in quad cells) is employed by Presagis’s Common Database (CDB) API [Pre14]. In order to reduce the differences in shapes and sizes between quadrilateral cells in the lat/long representation, different scales of longitude degrees are used to increase cell size near the poles. Five zones are defined over the Earth, with the interval between lines of longitude increasing from 1 to 6 depending on proximity to the poles.

C-squares (the Concise Spatial Query and Representation System) also discretizes the Earth using a lat/long parametrization [Ree03]. The domain underlying the parametrization is divided into four segments (NE, SE, SW, NW). Each cell then receives an index of the form $i_{xx}$; where $i$ is 1, 3, 5, or 7 if the cell is, respectively, located in the NE, SE, SW, or NW segment; $y$ is the first digit of the cell’s latitude; and $xx$ are the first two digits of the cell’s longitude. Note that, as the Earth is discretized once, multi-resolution is not provided.

As noted in [CR11, BZ07], Google Earth uses ClipMaps, also known as Universal Textures, which were proposed by Tanner et al. in [TM98] and patented in [Tan03]. Clip-Maps are modified versions of mipmap, which are collections of correlated images whose resolutions are decreased in a pyramidal structure (see Figure 30) [Wil83]. Clip-Maps are partial mipmap whose dimensions can be clipped to a maximum “clip size”, altering the pyramidal shape of the mipmap into one more closely resembling an obelisk (see Figure 30). This clipping is employed to reduce the size of the textures to a finite amount of memory in order to support rendering in real-time. This type of image-base representation of the Earth with an underlying 2D domain has been also used in Bing Map, Skyline Globe, and Nasa Wrold Wind [Sch14, sky, NAS]. For example, Bing Maps represents the Earth similarly, and employs a hierarchy-based indexing method based on quadtrees [Gar82] (see Figure 31).

While Google Earth uses a simple cylindrical projection [Goo07], Google Maps uses a Mercator projection to flatten the Earth onto a 2D map [Goo14]. The coarsest map resolution in Google Maps is an image tile with 256 × 256 cells, with finer resolutions obtained by dividing the tile into...
4. Data Types

Various geospatial data sets should be visualized, analyzed, and combined on a Digital Earth. These data sets are present in different types and formats and are obtained through several sources. In [VTP15], a variety of different geospatial data sets and their file formats are collected. In the following, we present some of the important geospatial data sets. Each category of data sets presented in the following is a field of research by itself and is studied in one or several review papers. Here, we present the important related works and survey papers that study these data sets and their related applications more comprehensively.

4.1. Elevation Data Sets

Digital Elevation Model (DEM) or Digital Terrain models (DTM) that capture the elevation of the surface of the Earth are typically provided by a set of regular grid points with an additional height (Figure 33 (a)). As well as grid based representation of the elevation data, Triangulated Irregular Networks (TIN) are also used in which the elevation model is represented on a set of triangular faces with arbitrary connectivity (Figure 33 (b)). In order to use these data sets in Digital Earth frameworks, they are usually converted to a cell based representation by considering a height attribute for each cell (Figure 33 (c)). Rendering, compression, multiresolution representation of terrain models in both TIN and regular grid formats are very well studied and survey papers exist that discuss these subjects related to terrains [DFM’06, DFP92, PG07, WDF11]. In addition to the existing elevation data sets, terrain models can be also synthesized through interactive techniques [ZSTR07, vBBK08, TEC’14] or by adding details using fractals or Perling noise functions [SBW06, Ebe03]. These synthetic elevation data sets are usually used for games, simulation, or aesthetic designs and do not have a use in real geospatial data analysis.

4.2. Imagery Data sets

Geospatial imagery data sets are very useful for visualization and analysis of locations. These data sets are typically categorized as aerial or satellite photographs. In the first, images are taken by platforms such as aircraft, helicopters, balloons, etc that do not have a fixed support on the Earth while the later is taken by a satellite. These images are usually used as textures for the cells of the Earth at multiple resolutions, analyzing its relevant data, or providing interesting views such as spherical panoramic views. Multiresolution representation of a single image is a very well studied subject in image processing [SDS96] and analyzing geospatial data sets are usually studied remote sensing digital image analysis [RR99, TCV09, TL10]. However, in Digital Earths, multiple images at different scales exist and providing a meaningful connection between these images is an important subject of research [ZAG14] (see Figure 34 (a) and (b)). In [HH10], a continuous transition between correlated images at different resolutions is established by using a smooth mipmap pyramid structure for multiple images. In [LF11], a smooth transition is proposed for large scale changes in multiresolution images by blurring the boundary of high and low resolution images. Spherical panoramic views of geospatial locations are also very interesting and useful (Figure 34 (c)). Although creating spherical panoramic views of a set of given images is a well studied subject [ZC04], nicely stitching the images [CBK’11, LHL’11], smoothly navigating through different views [ZW+F’13] and distortion correction for the spherical mappings [KH10, GSHD09] are some examples of interesting current problems in generating spherical panorama.
4.3. Vector Data

Vector data sets are very important in the visualization of Digital Earths. These data sets are typically represented in points, polylines, or polygons representing, the boundaries of regions (e.g., countries, cities), roads, and rivers. Visualizing vector data sets on terrains have been extensively studied [SK07, SGK05, VTW11]. However, combining vector data sets on Digital Earths which has a multiresolution representation of a series of data sets is still a challenge. As described in [ZCG15], visualization of vector data sets on 3D terrains can be categorized into three groups: texture-based, geometry-based, and shadow volume-based approaches (see Figure 35). In texture-based approaches, vector data is rasterized into textures and mapped onto the terrain surface [KD02]. In geometry-based approach, the vector data has its own separate geometry from its underlying terrain but the geometry is slightly modified if needed to be adapted with the terrain [WKW*03]. This method of vector data representation has been extensively used when both digital elevation data sets and vector data sets need to be represented simultaneously [SGK05, SBZ07, ARJ06, QWS*11, WSFL10]. In [SK07], a shadow volume-based approach is presented in which the vector geometry is extruded into polyhedrons that are later rendered into the stencil buffer to generate an appropriate mask. This mask is correspondent to the projection of the vector data onto the terrain surface and distinguishes visible parts from invisible parts of the scene in the rendering process. Visualizing Vector data sets are important part of Digital Earths and most of the available Digital Earths support these data sets in their frameworks [NBC10, OGW*13]. In [MW A*13], various methods and techniques for reconstruction of 3D geometric models have been studied. Many of available Digital Earths have not included 3D geometric data sets in their frameworks. Google Earth however, has nicely incorporated these entities in its framework (see Figure 37).

4.4. 3D Geometric Data Sets

Another set of data sets that can be added to Digital Earths frameworks are 3D geometric data sets [NBC10, OGW*13]. These data sets include 3D buildings, houses, towers, bridges, cars or trees. Such 3D models are usually made by interactive techniques or using a large set of images or Lidar data sets that include a large amount of point clouds [HDT*07, AFS*11, NBC10]. In [MWA*13], various methods and techniques for reconstruction of 3D geometric models have been studied. Many of available Digital Earths have not included 3D geometric data sets in their frameworks. Google Earth however, has nicely incorporated these entities in its framework (see Figure 37).

5. Applications

The future of the Digital Earth framework and its applicability to different use cases have been the subject of extensive discussion and anticipation [But06, TAH08, SWK, GC07, CdBJ*12, GGA*12, LYSW14, CdBJ*12, ACE*11, Goo08, Goo12]. Thus far, a number of these insights and visions
have been realized in applications for a variety of different fields. Facilitating the search and visualization of media resources stored on social networks [BAG*13], automatic quality control for 3D reconstructions [ZZLH07], sharing geological data sets through a common framework [TGC*14, WJGF*11], and navigation and traffic simulation [SWL11, WSL12, DP09] are just some of the applications for which the Digital Earth has been employed.

Google Earth, in particular, has seen a lot of use in such applications as the environment is user-friendly and many of its features are freely available.

5.1. Environmental Applications

One of the main functions of the Digital Earth framework is environmental data set analysis, which can be used for the monitoring of landscape and environmental changes, weather prediction, disaster prediction, and endangered animal control. These tasks have traditionally posed a challenge due to the dynamic, massive, complex, and versatile nature of environmental data sets [WKO92, Wil11, LJY*13, LWY*11, JRP08].

5.1.1. Environment Monitoring

Observing, visualizing and monitoring environmental data sets are difficult tasks that may be significantly simplified using the common reference model provided by a Digital Earth. Erickson, Michalak, and Lin employed Google Earth to visualize atmospheric CO₂ models and showed that the Digital Earth can be used to familiarize both the general public and decision makers with scientific concepts that describe the Earth’s systems [EML10]. Climate change in particular has provided the impetus for many works in this area, especially those that monitor and visualize changes in snow, ice, and glacier data sets [She05, BPS07, LKO11, BRD*11a, SHS07], which have important ramifications not only for the environment and climate but also on the tourism and fishing industries. Additional applications include the monitoring of water quality [JTW07, BRD*11b], cropland area estimation [LG13], and visualization of geochemical rock and sediment data sets [YST*11].

5.1.2. Disaster Control

An important component of climate monitoring is the visualization and analysis of data sets relevant to modeling or predicting natural disasters, for which the Digital Earth may be used as a source of data and common reference framework [KBG13]. For instance, although modeling and visualizing tsunamis is generally hard, the Digital Earth has proven helpful [ZSY*08, YNL*06] due to the relative ease of visualizing and integrating different data sets. The same can be said for earthquake visualization [KBB*08], real-time fire alert system design [DVVF08], and forecasting of flash flooding [BF09].

5.1.3. Endangered Species Monitoring

An important consequence of environmental and climate changes are their impacts on plant and animal species. The Digital Earth provides a cheap, simple, and generally accurate means through which animal [Web14] and plant [BN09, PMG*13, Rei09] diversity and density may be monitored and landscapes that are not easily accessible may be studied [GDMGC*14]. Google Earth has been used in this area to study and monitor species habitats [BBD*11, OMT13] and marine life [SCT*10, SAP11, MMB*13].

5.2. Health

Though not as strongly affected by environmental changes as animals and plants, humankind continues to be plagued by diseases throughout the globe. The visualization and management of disease-related data sets on a Digital Earth can prove useful to the analysis of the causes and spread of diseases and their resistance to drugs [BSS11, SSU*09]. Using Google Earth, a low-cost surveillance system for Dengue fever was developed in [CPJ*09], for the spread of avian influenza (H5N1) in [JHG*07], and for injury data collection in low income countries in [CS10]. In addition, Zhang, Shi, and Zhang used Google Maps to recognize patterns in data sets featuring epidemiological and geographical information [ZSZ09].

5.3. Urban Design

The benefit of a hierarchical reference model in a Digital Earth is the possibility to model geospatial data at smaller scales than that of the entire globe, such as that of a city. 3D city models are digital models that feature buildings, terrains, vegetation, and other elements of urban areas [For99]. Owing to the relative ease of integrating and visualizing geospatial data within these frameworks and their usefulness in the distribution, access, manipulation, and presentation of massive location-based data sets [Ros10], the Digital Earth provides a handy infrastructure for the modeling of 3D cities. In [WKBK10], a Digital Earth is used to visualize a 3D city model in order to raise awareness of energy sources and changes in energy consumption within an urban environment, and in [WHG10] is used to support public participation in urban planning projects.

5.4. Education

Beyond facilitating data set visualization and processing for particular areas of study, the Digital Earth can also provide opportunities for educators to provide an interactive environment for learners to understand spatial, geographical, and environmental concepts [SKP08]. A survey of this framework as applied to teaching applications can be found in [ROH08], with a focus on Google Earth, NASA World Wind, Microsoft Virtual Earth (succeeded by Bing Maps),
and Skyline Globe. In addition to Google Earth, other Digital Earth software, such as PYXIS’s WorldView [Lan15], have been recently integrated into pedagogical practices.

There have been studies that show that using the Digital Earth in learning environments can have a significant positive impact on a student’s understanding of spatial concepts. Thankachan and Franklin [TF13] studied what happened to 102 sixth grade students’ average social studies grade when Google Earth was incorporated into the content of the class, and observed that students with access to Google Earth obtained a better average grade. The effectiveness of Google Earth has been also examined in secondary school geography lessons [DKK13], indicating that using an interactive environment like Google Earth can provide an engaging environment that helps students learn geographical concepts. Similar results have been observed in the learning environments of environmental science, GIS, and urban studies [BAK14, HARR12, Pat07, Bod08].

6. Conclusion and Future Work

Although there exists a huge amount of work related to Digital Earth and DGGSs, there are still many unanswered questions and challenges both in fundamental and application aspects. These challenges mostly include big data representation, creative visualization for geospatial data sets, handling moving and unpredictable data sets such environmental data sets and many more. Having an efficient Digital Earth as a reference model can be a useful framework on which all of these data sets can be combined, analyzed, and visualized.

We believe that this paper provides a deep overview about the construction of Digital Earths and its discussion about current working Digital Earths can be beneficial for interested researchers and engineers. In addition, key research areas in computer graphics and visualization community that are presented here can be useful for people with backgrounds in GIS, and Geomatics that are interested to benefit from state-of-the-art computer graphics techniques in Earth representation applications.

References

[AMA14] ALI MAHDAVI-AMIRI / A Survey of Digital Earth Representation and Visualization


FF90 Fekete G., Treinish L. A.: Sphere quadtrees: a new data structure to support the visualization of spherically distributed data. 242–253. 2, 6

Gar82 Gargantini I.: An effective way to represent quadtrees. Commun. ACM 25, 12 (1982), 905–910. 6, 10


[KPKPS89] Kamgar-Parsi B., Kamgar-Parsi B., Sander
Ali Mahdavi-Amiri / A Survey of Digital Earth Representation and Visualization


Handling, International Geographical *Proceedings of the 8th International Symposium on Spatial Data Processing units (gpus) and multi-core central processing units in visualizing 3d/4d environmental data using many-core graphics processing units (cpus).*


Harnessing the Power of Google Earth for Seagrass Conservation in the Comoros Islands. *Oceanography 16*, 1 (2003), 11–19. 10

The use of satellite imagery for the identification and protection of important oak areas. In *First International Conference on Remote Sensing and Geoinformation of Environment (2013), International Society for Optics and Photonics*, pp. 879522–879522. 13


Tiles à la google maps: Coordinates, tile bounds and projection. 2008. 11


A lattice-based indexing system and its applicability to the description of oceanographic datasets. *Oceanography 16*, 1 (2003), 11–19. 10


Ros D.-I. L.: Virtual 3d city models in urban land management. 13


Rosca D., Plonka G.: An area preserving projection from the regular octahedron to the sphere. *Results in Mathematics 62*, 3-4 (2012), 429–444. 5

Nasa World Wind. http://worldwind.arc.nasa.gov/java/, 10


P088  PRIDAL K. P.: Tiles à la google maps: Coordinates, tile bounds and projection. 2008. 11


NAS  NASA World Wind. http://worldwind.arc.nasa.gov/java/, 10
SWK: Sahr K., White D., Kimerling A. J.: Geodesic discrete global grid systems. 1, 2, 6, 12.
TL10: Tieh D., Lang S.: Analytical 3d views and virtual globesâ€”scientific results in a familiar spatial context. ISPRS


Indexing the aperture 3 hexagonal discrete global grid. J. Vis. Comun. Image Represent. 17 2, 3, 6, 7, 9


High-quality cartographic roads on high-resolution dems. 12
