

Chapter 2

Digital Earth Platforms



**Troy Alderson, Matthew Purss, Xiaoping Du, Ali Mahdavi-Amiri
and Faramarz Samavati**

Abstract In this chapter, we provide a thorough discussion on Digital Earth with particular focus on Discrete Global Grid Systems (DGGS), which are a standardized representation of the Earth. We describe the necessary components of a DGGS, such as the underlying 2D representation, indexing system, projection, and cell types. We also discuss a selection of well-known public and commercial DGGSs followed by current DGGS standards.

Keywords Discrete Global Grid · OGC Standard · Digital Earth

2.1 Introduction

Digital Earth is a framework for geospatial data management. In this model, data are assigned to locations on a 3D model of the Earth and analyzed at multiple resolutions, each representing data at a specific level of detail. To locate and retrieve data sets associated with the Earth, mechanisms are needed for data representation, region addressing, and the assignment and retrieval of data for a region of interest. Digital Earth provides a reference model that can handle all of these queries.

T. Alderson · F. Samavati
Department of Computer Science, University of Calgary, Calgary, Canada
e-mail: tfalders@ucalgary.ca

F. Samavati
e-mail: samavati@ucalgary.ca

M. Purss
Geoscience Australia, Canberra, Australia

X. Du
Key Laboratory of Digital Earth, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, China

A. Mahdavi-Amiri (✉)
Department of Computing Science, Simon Fraser University, Burnaby, Canada
e-mail: amahdavi@sfu.ca

Two main approaches have arisen for the creation of a Digital Earth: Discrete Global Grid Systems (or DGGSs for short), and datacubes. In a DGGS, the surface of the Earth is discretized into a set of highly regular spherical/ellipsoidal cells. These cells are then addressed using a data structure or indexing mechanism that is used to assign and retrieve data. Datacubes are n-dimensional arrays that store geospatial data, ordered according to various attribute/coordinate axes, which can be spatial or non-spatial in nature.

In this chapter, we focus particularly on Discrete Global Grid Systems, which have global scope, more readily support interoperability than datacubes, are generally compatible with conventional datacube approaches, and can be used to provide the back-end support for a datacube implementation (Purss et al. 2019). The following section discusses DGGS as well as its components and their characteristics in detail.

2.2 Discrete Global Grid Systems

The traditional approach to discretizing the Earth is to use a latitude/longitude coordinate system on a sphere (Cozzi and Ring 2011), in which the 2D domain (or planar map of the Earth) is partitioned into a grid of cells by discretizing the 2D latitude/longitude domain. These cells may be further subdivided to increase the resolution and mapped to the sphere through the use of spherical coordinate equations and/or an appropriate spatial projection. The resulting spherical cells are primarily quadrilateral, though singularities and triangular cells appear at the poles, and the areas of the cells vary according to the latitude.

In order to better represent the Earth with more uniform cell structures, a polyhedron with the same topology as the sphere can be used. With an initial discretization of the Earth into planar cells (typically produced by considering the planar faces of an approximating polyhedron), the initial cells may then be refined to an arbitrary resolution and mapped from planar cells to spherical cells via some projection method (see Fig. 2.1). Given a regular refinement, a multiresolution hierarchy between cells (in which each cell has a coarser resolution parent and a number of finer resolution

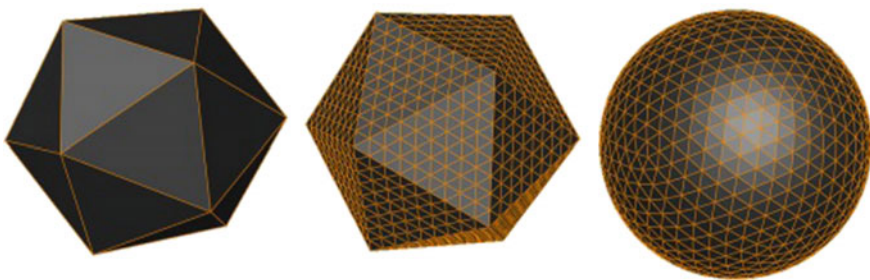


Fig. 2.1 An initial polyhedron that has been refined and projected onto the sphere

children) with a semiregular cell structure can be systematically defined. To query and render cells, they are then typically addressed via an indexing mechanism.

DGGSs are defined in terms of several different components. The main components of a DGGS include the initial planar or piecewise domain, cell type, projection, indexing, and refinement. In the following, we discuss each of these components in more detail in the context of DGGS construction.

2.2.1 Initial Domain

The simplest domain for a DGGS is a 2D map of the Earth, of which latitude/longitude grids are the standard. As these tend to exhibit large distortions across the globe and singularities at the poles, complicating queries and data analysis, a spherical polyhedron can instead be used for the initial domain of a DGGS. Such DGGSs are also known as Geodesic DGGSs (Sahr et al. 2003).

The most common choices for an initial polyhedron are the platonic solids—the tetrahedron, cube, octahedron, dodecahedron, and icosahedron—in addition to the truncated icosahedron. Each of these polyhedra offers distinct benefits. For instance, the octahedron defines a simple and symmetric domain, and can be unfolded into a very simple quadrilateral domain. Cubes are made of quadrilateral facets that are appropriate for the generation of quadrilateral cells. In comparison to other polyhedra, the icosahedron and truncated icosahedron undergo less angular distortion when processed through an equal area projection (Snyder 1992).

2.2.2 Cell Type

There are three main cell types that are used in a DGGS: hexagonal, triangular, and quadrilateral. If the initial domain is a polyhedron, other extraordinary cell types may be present, such as pentagons in a truncated icosahedron. However, the number of extraordinary cells is fixed no matter the resolution. Each of the three main cell types presents some advantages over the others that ought to be considered when selecting the base cell of a DGGS. For instance, quadrilateral cells are congruent, compatible with Cartesian coordinate systems, easy to index, and compatible with standard rendering libraries. Triangular cells are planar, easy to use, compatible with standard rendering techniques and libraries, and congruent. Hexagonal cells are the best for sampling, with the smallest quantization error, and they have uniform adjacency. Depending on the initial domain of a DGGS and the application that the DGGS was designed for, any of these types of cells can be employed as the DGGS cell type.

2.2.3 Refinement

Refinements are used to produce finer cells from an initial set of coarse cells. In a DGGs, they can be used to construct cells at multiple resolutions on the sphere by refining the faces of a polyhedron. Refinements are in part characterized by a ratio known as the aperture, or factor, of the refinement. This ratio relates the number of coarse cells to the number of fine cells at the next resolution, and several different apertures have been employed in DGGs. After applying a refinement, the resulting fine cells may be assigned to a coarse *parent* cell as *children* of that cell, producing a hierarchical structure that is useful for many grid and spatial processing operations. Traversing from a parent cell to its children or from a child to its parent is known as hierarchical traversal.

In addition to the factor of refinement, other aspects play an important role in characterizing a refinement, such as congruency and alignment. In a congruent refinement, a coarse cell encompasses precisely the same area as a union of finer cells at the next resolution (see Fig. 2.2). For example, the 1-to-4 quadrilateral refinement shown in Fig. 2.2a is congruent while the 1-to-3 hexagonal refinement shown in Fig. 2.3b is not. Assigning a set of fine cells to serve as the children of a coarse cell is trivial in congruent refinements, as children are uniquely covered by a single parent cell and, therefore, the handling of hierarchical traversal queries is simplified. In contrast,

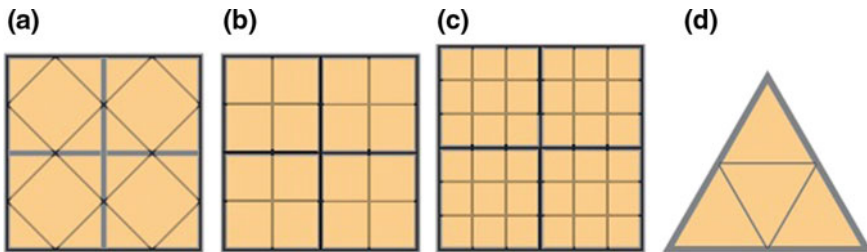


Fig. 2.2 (a) A center-aligned 1-to-2 refinement that is not congruent. (b) A vertex-aligned and congruent 1-to-4 refinement. (c) A center-aligned and congruent 1-to-9 refinement. (d) A center-aligned and congruent 1-to-4 refinement for triangles. The boundaries of coarser cells are highlighted using thicker lines

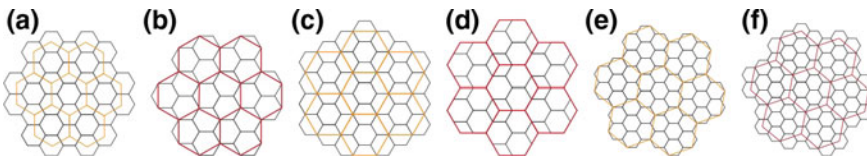


Fig. 2.3 (a) Center-aligned and (b) vertex-aligned 1-to-3 refinements. (c) Center-aligned and (d) vertex-aligned 1-to-4 refinements. (e) Center-aligned and (f) vertex-aligned 1-to-7 refinements. Note that each of these refinements is incongruent

incongruent refinements create ambiguity in the assignment of a child to a coarse cell, since there may exist several potential parents for a fine cell (Fig. 2.3).

If, after applying a refinement, every coarse cell shares its centroid with a fine cell, then that refinement is called center-aligned. Any DGGs that employ such refinements are likewise called center-aligned. If such a property does not hold for the refinement, then it is usually vertex-aligned, meaning that the parent and child cells share a vertex (Fig. 2.2).

Various types of refinements exist for quadrilateral, triangular, and hexagonal DGGs cells. However, whereas many different refinements have been employed in DGGs based on quadrilateral and hexagonal cells, DGGs represented using triangular cells generally use 1-to-4 refinements (see Fig. 2.2d). Quadrilateral refinements can be congruent whilst hexagonal refinements never are (see Fig. 2.3). Consequently, parent-child relationships must always be explicitly defined in a hexagonal DGGs. However, once defined, this becomes a static feature of the DGGs infrastructure, allowing for consistent hierarchical traversal of the grid system, and incongruent refinements still exhibit characteristics that can be useful in a DGGs. Hexagonal 1-to-3 refinement has the lowest aperture of all hexagonal refinements; while hexagonal 1-to-4 refinement produces rotation-free lattices at all levels of resolution (simplifying hierarchical analysis in contrast to other refinements). Of the refinements shown in Fig. 2.3, fine cells in the 1-to-7 refinement cover the hexagonal coarse shape better than other refinements, and therefore more closely resemble congruency and provide a simpler hierarchical relationship between the cells. As a result, there is growing interest in this type of refinement (Middleton and Sivaswamy 2005).

2.2.4 Projection

Projections have traditionally been used to create maps of the Earth. Various forms of projection can be used to flatten the Earth (usually treated as spherical), and these can be categorized into different types, such as conformal, gnomonic, or equal area (Grafarend et al. 2014). When a spherical projection is used, some unavoidable distortions appear that one may try to reduce. In the following, we discuss several spherical projections in more detail.

2.2.4.1 Traditional Cartography

In traditional cartography, a spherical projection is a transformation from a point on the Earth (a sphere) to a point on a 2D map. Such a projection can be represented as a function: $P' = F^{-1}(P)$, where P lies on the sphere, and P' lies on the 2D map (see Fig. 2.4). The simplest method that can be used to create a 2D map from a spherical representation of the Earth is to use spherical coordinates $((\theta, \varphi))$. Doing so involves cutting the Earth (e.g. along a meridian) and unfolding it to form a 2D square, with θ (the longitude) and φ (the latitude) serving as the two main axes of

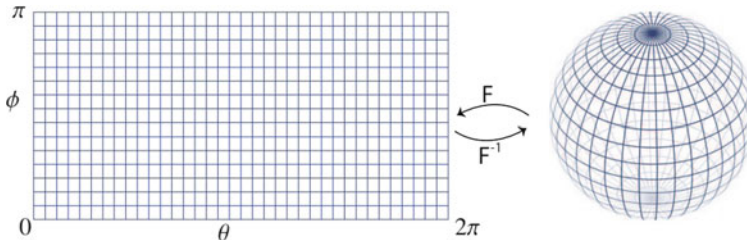


Fig. 2.4 2D domain and its corresponding sphere

the 2D domain. This 2D domain and its corresponding sphere are related through the following equations:

$$F(\theta, \varphi) = \begin{pmatrix} R \cos(\theta) \sin(\varphi) \\ R \sin(\theta) \sin(\varphi) \\ R \cos(\theta) \end{pmatrix}$$

$$F^{-1}(x, y, z) = \begin{pmatrix} \theta \\ \varphi \end{pmatrix} = \begin{pmatrix} \tan^{-1}\left(\frac{y}{x}\right) \\ \cos^{-1}\left(\frac{z}{R}\right) \end{pmatrix}$$

where R is the radius of the sphere: $R = \sqrt{x^2 + y^2 + z^2}$.

As can be observed from Fig. 2.4, perfect squares on the 2D domain are mapped to distorted quadrilateral or even triangular cells (at the poles) on the sphere. In order to reduce different distortions, it is possible to define alternative mappings, which can be equal-area, conformal (angle-preserving), or possibly stretch-preserving. Since the purpose of a DGGS is to use planar or piece-wise planar (i.e. polyhedral) domains to sample the spherical surface of the Earth, the use of an equal-area projection (in which data values sampled from the Earth occupy similar areas in the DGGS) is often desired (White et al. 1992). Here, we describe some of the most commonly used equal-area projections. There are several traditional options, such as the projections proposed by Lambert (cylindrical and azimuthal), Mollweide, and Werner (see Fig. 2.5) (Snyder 1992).

Among these projections, the Lambert azimuthal equal-area projection is particularly interesting, since it serves as the base projection for a number of equal-area polyhedral projections. In this projection, a point P on the sphere S is projected to a point P' on the 2D domain (2D map). To find P' , a plane ρ which is tangent to S at another point C is used. Then, P' is the intersection of ρ with a circle that has its origin at C , passes through P , and is perpendicular to ρ (see Fig. 2.6). Note that the antipode of C (i.e. the point diametrically opposite C) is excluded from the projection as its intersecting circle is not unique. In addition, C is projected to itself along a circle of radius 0. This projection and its inverse can be explicitly described using simple mappings:

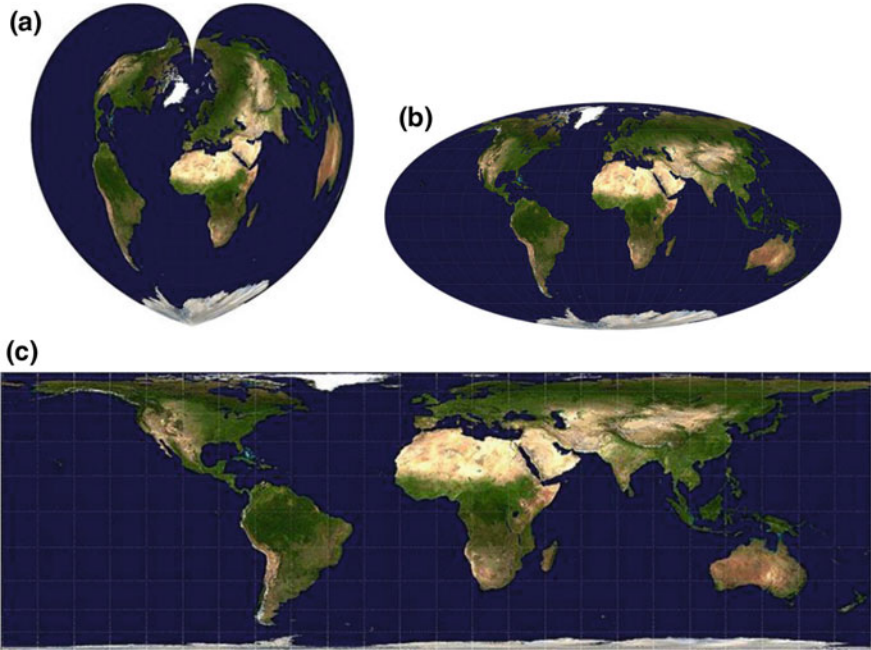


Fig. 2.5 (a) Werner, (b) Mollweide, and (c) Lambert (cylindrical) projections

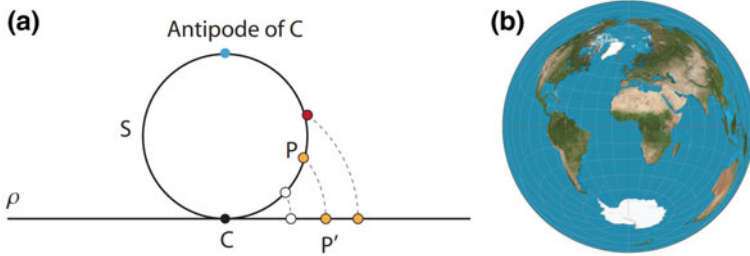


Fig. 2.6 (a) Lambert projection from sphere S to plane ρ . (b) 2D domain of the Earth resulting from Lambert Azimuthal equal-area projection. The second image is taken from Wikipedia

$$F(x, y, z) = \left(\sqrt{\frac{2}{1-z}}x, \sqrt{\frac{2}{1-z}}y \right),$$

$$F^{-1} = \left(\sqrt{1 - \frac{X^2 + Y^2}{4}}X, \sqrt{1 - \frac{X^2 + Y^2}{4}}Y, -1 + \frac{X^2 + Y^2}{2} \right).$$

2.2.4.2 Projection for Polyhedral Globes

A projection defined for a given polyhedron is characterized by a function F that maps each point P on the sphere to a face of the polyhedron. Consequently F^{-1} is defined as a function that maps points on the polyhedron to the sphere. Often, in order to construct these projections, a traditional projection defined between the sphere and a plane is used individually on each face. For instance, Snyder's equal-area projection (which is commonly used in DGGs) employs Lambert azimuthal equal-area projection individually for each face. The problem is reduced to projection for right triangles by splitting the faces of the polyhedron along lines of symmetry (see Fig. 2.7). A scaling factor is then found between the radii of two spheres: one with the same area as the (spherical) polyhedron, and another that circumscribes the polyhedron. Finally, a triangle on the polyhedron (whose area matches that of the corresponding spherical triangle that encompasses point P) is generated. The final equation for Snyder's projection is presented in closed form as:

$$x = \rho \sin(Az'),$$

$$y = \rho \cos(Az'),$$

in which the ratio ρ and angle Az' are defined in terms of a set of trigonometric functions and known constants that are provided in (Snyder 1992).

While the forward form of the Snyder projection is presented in a simple closed form, its inverse calculation requires finding the roots of a nonlinear equation (Snyder 1992). Snyder suggests the use of the Newton-Raphson iterative technique to compute the inverse projection, which can slow down the process of mapping points from the polyhedron to the sphere. To reduce this inefficiency, Harrison et al. (Harrison et al. 2011, 2012) worked to optimize the inverse Snyder projection by providing initial estimates to the Newton-Raphson technique that are close to the roots of the nonlinear

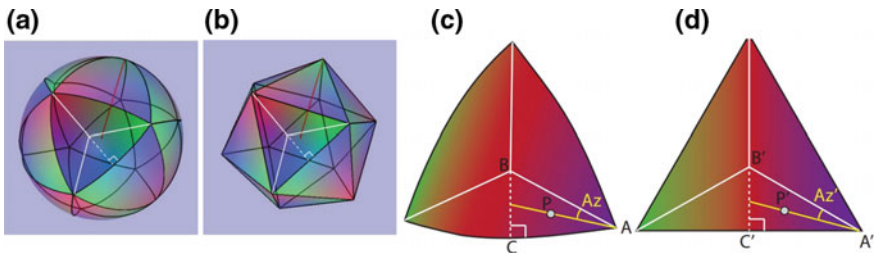


Fig. 2.7 An icosahedron (a) after and (b) before projection to the sphere. (c) The projection operates on right triangles by splitting the initial triangles. Each spherical triangle is associated with a triangle of the polyhedron. Each point P in (c) is associated with some point \hat{p} in (d). Az' is the angle between A and the great circle arc passing through P and A

equation. These initial estimates are found using a polynomial curve that fits the roots of the non-linear equation.

In addition to Snyder’s projection, other types of equal-area projection have also been used in DGGs. For instance, Roşca and Plonka’s projection (Roşca and Plonka 2011) was used in the one-to-two Digital Earth, which is a cube-based Digital Earth (Mahdavi-Amiri et al. 2013), and extended to the octahedron in (Roşca and Plonka 2012). This equal-area projection describes a mapping from a cubic domain Ω to a spherical domain Δ . The main idea behind this projection is to map each face f of Ω onto a partition of Δ . To this end, an intermediate domain, called a curved square, is used. As a result, the projection involves two steps. First, f is mapped to a curved square on the tangent plane of Δ , parallel to f , using an equal-area bijection denoted by T . Then, the curved square is mapped to a partition of Δ using inverse Lambert azimuthal equal-area projection, denoted by F^{-1} (Fig. 2.8).

HEALPix (see Fig. 2.9) is also a cube-based equal-area projection, and is a hybrid projection of Lambert cylindrical equal-area projection and Collignon pseudo-cylindrical equal-area projection (Gorski et al. 2005). Lambert cylindrical projection is used for equatorial regions of the Earth while Collignon is used for the polar regions.

These equal-area projections are not the only projections that can be used in a DGGs but are examples of projections that have been used already. Naturally, a DGGs designer should always use a projection suited to the needs of their application.

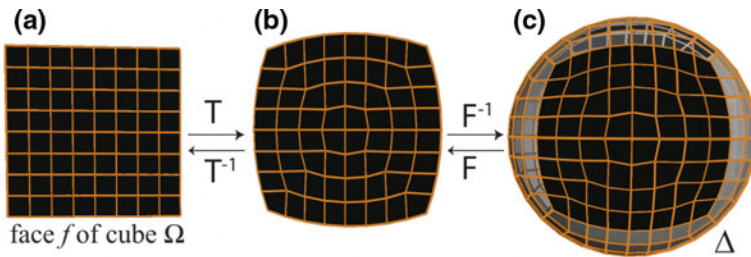


Fig. 2.8 Steps of the spherical projection. Points on a face f of Ω (a) are projected onto a curved square (b) and then projected onto a portion of the unit sphere Δ (c)

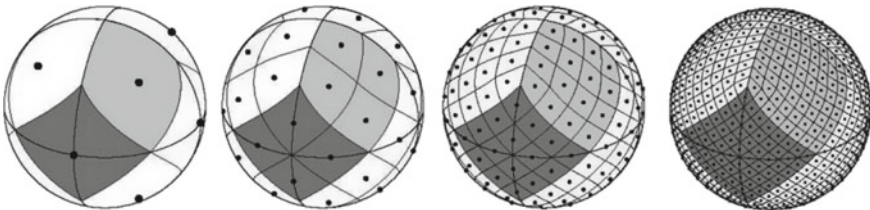


Fig. 2.9 HEALPix projection at four successive resolutions

2.2.5 Indexing

In order for a Digital Earth to handle queries related to location-based data efficiently, a hierarchical approach to data storage is needed. Hierarchical data structures such as quadtrees have been used in various Digital Earth frameworks (Fekete and Treinish 1990; Tobler and Chen 1986), but are typically shelved in favor of indexing methods in order to avoid the cost of expensive tree structures that record node dependencies. Given an indexing method for a DGGs, the method must ensure that, at each resolution, each cell receives an index that uniquely identifies the cell. This index may then be used with reference to a data structure or database in order to retrieve data associated with the cell.

Although various types of methods exist to index the cells of a DGGs, they are typically derived from three types of general indexing mechanisms: hierarchy-based, space-filling curve-based, and axes-based. In the following, we describe each category and provide some examples.

2.2.5.1 Hierarchy-Based Indexing

Applying refinements to a polyhedron produces a hierarchy that can be used to index cells. When a refinement is applied to a set of coarse cells, fine cells are created and assigned to coarse cells through a parent-child relationship. It is possible to use this relationship to define an indexing system by assigning an initial index to each cell at the first (i.e. lowest) resolution, and then using each cell's index as a prefix to the indices of its children. Formally, if a coarse cell has index $I d_0 d_1 \dots d_{r-1}$, then its children receive indices of the form $I d_0 d_1 \dots d_{r-1} d_r$, where d_r is an integer whose range is known as the base of the indexing method, denoted by b (i.e. $d_i \in [0, b-1]$).

The base is used to define algebraic operations on indices, such as conversion to and from the Cartesian coordinate system, or neighborhood finding (Tobler and Chen 1986; Vince and Zheng 2009). Hierarchy-based indexing is very efficient in supporting hierarchical queries, although neighborhood-finding tasks may require complex algorithms, depending on the base of the indexing method and the algebra defined for the indexing system. An example of this type of indexing was proposed in (Gargantin 1982) for quadrilateral cells resulting from 1-to-4 refinement (see Fig. 2.10). Here, the children resulting from 1-to-4 refinement on a quadrilateral with index I receive indices I_0, I_1, I_2, I_3 .

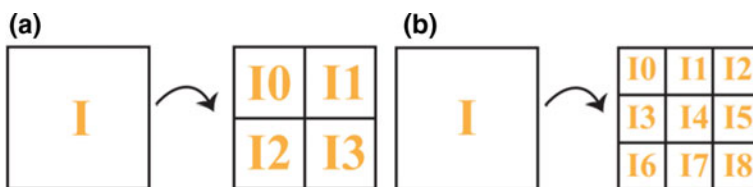


Fig. 2.10 Hierarchy-based indexing systems in (a) base four and (b) base nine

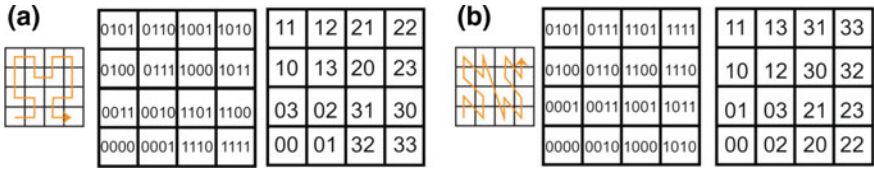


Fig. 2.11 (a) (Left) Hilbert SFC. (Middle) Indexing in base two. (Right) Indexing in base four. (b) (Left) Morton SFC. (Middle) Indexing in base two. (Right) Indexing in base four

2.2.5.2 Space-Filling Curve Indexing

Another method for indexing cells in a DGGS is to use a space-filling curve (SFC) as a reference for the indexing (Mahdavi-Amiri et al. 2015b). SFCs have been used in many applications, such as compression, rendering, and database management, and are 1D curves (often recursively defined) that cover a particular space.

Recursively defined SFCs start from a simple initial geometry defined on a simple domain (usually a square). The domain is then refined and the simple geometry is repetitively transformed to cover the entire refined domain. Typically, if the initial geometry covers i cells, a 1-to- i refinement is suitable for use in generating a refined domain. In this way, each SFC is associated with a refinement. To index cells based on an SFC, decimal numbers can be employed, though the corresponding indices do not directly encode the resolution. To resolve this issue and obtain a compact indexing, a base b for the indexing that is compatible with the refinement is often chosen. With a 1-to- i refinement, this usually means that the base of the indexing method is taken to be i or \sqrt{i} (see Fig. 2.11). For instance, the refinement associated with the Hilbert and Morton curves is 1-to-4. Therefore, when using a Hilbert and Morton curve, a base four or base two indexing is appropriate.

Indexing methods derived from SFCs have been widely used in DGGS and terrain rendering. For instance, in (Bai et al. 2005; White 2000), Morton indexing was used to index cells resulting from 1-to-4 refinements on the icosahedron and octahedron, while in (Bartholdi III and Goldsman 2001), the Sierpinski SFC was used to index triangular cells refined with a factor of two.

2.2.5.3 Axes-Based Indexing

Another mechanism for indexing is to use a coordinate system with m axes, U_1 to U_m , that span the entire space on which the cells lie. Then the cells' indices can be expressed as m -dimensional vectors (i_1, i_2, \dots, i_m) , in which the i_j are integer values that indicate the number of unit steps taken along each axis, U_j . If, alternatively, a 1D

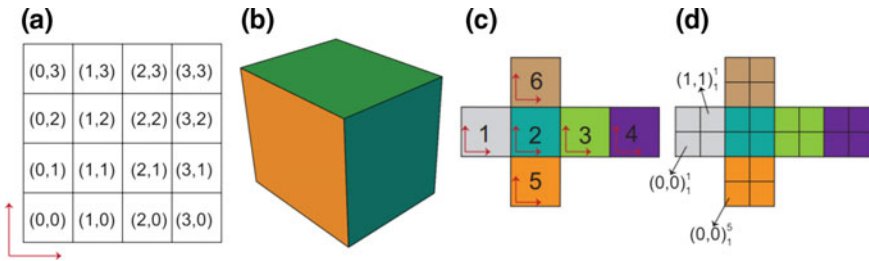


Fig. 2.12 (a) Integer indexing of cells using Cartesian coordinates. (b) A cube. (c) The unfolded cube in (b) and coordinate systems for each face. (d) Indices of some cells after one step of 1-to-4 refinement

index is required, these integer values can be appended together in a string, separated using a delimiter character. A simple example of such an indexing can be used to index a quadrilateral domain with Cartesian coordinates, as illustrated in Fig. 2.12a. When a refinement is applied to the cells, a subscript r is appended to the index in order to encode the resolution. In the proposed axes-based indexing methods for DGGs, m is typically taken to be two or three. For instance, 3D indexing was used in (Vince 2006) by taking the barycentric coordinate of each cell to be its index.

A 2D indexing method can be applied on the polyhedron used to construct a DGGs by embedding the polyhedron’s faces into a 2D domain and defining a coordinate system over that domain (Mahdavi-Amiri and Samavati 2012). In this way, each face can be given its own coordinate system (Mahdavi-Amiri et al. 2013, 2015a; Mahdavi-Amiri and Samavati 2014). Figure 2.12 illustrates an indexing for the quadrilateral cells of a cube after 1-to-4 refinement, where each face has its own coordinate system. In order to distinguish between the cells associated with each face, an additional component that refers to the initial polyhedral face can be added to the indices. For example, index $(a, b)_r^f$ refers to cell (a, b) in face f at resolution r (Fig. 2.12d).

2.2.5.4 Remarks on Categorization

Note that this categorization of index types is primarily intended to reflect the core idea used to construct the indexing methods and can be used to easily identify which operations can be handled naturally by a particular indexing system. For example, hierarchy-based indexing methods naturally lead to efficient hierarchical traversal operations. However, well-designed indexing methods must necessarily also consider other properties and support other operations. For example, it is certainly possible to handle neighborhood finding in hierarchy-based indexing methods, although not as efficiently or as naturally as with axes-based techniques. Based on the pattern of indices, some indexing methods can be interpreted as belonging to two categories (e.g. SFC or hierarchy-based). However, an indexing method is either constructed through use of a parametrized curve or it inherits the index of its parent. It is possible to use a parametrized SFC that indexes the children and prefixes the parent’s index.

This indexing method is considered to be SFC-based, since the construction of the indexing is based on the parametrization of the SFC and not on the hierarchy of the cells.

2.3 Scientific Digital Earths

Now that we have discussed the various components that define different DGGs, let us examine some specific DGG constructions that have been proposed in the literature. Note that some proposed DGGs are left for the following section, in which we survey some of the existing Digital Earth implementations.

Among the earliest proposed DGGs are those designed by Digital Earth pioneers M. Goodchild and G. Dutton. Goodchild’s HSDS (Hierarchical Spatial Data Structure (Goodchild and Shiren 1992)) is built upon an initial octahedron. Unlike many other DGGs, the faces of this octahedron are inverse projected to the sphere before the generation of finer cells, and the refinement (a congruent, 1-to-4 refinement) is applied directly on the resulting spherical triangles (using geodesic rather than Euclidean midpoints). The child cells are also spherical triangles, and are indexed using a hierarchical base-4 numbering scheme (see Fig. 2.13). Unfortunately, the projection implied by the refinement method is neither equal-area nor conformal.

Dutton’s QTM (Quaternary Triangle Mesh (Dutton 1999)) is also constructed using congruent 1-to-4 refinement on an initial octahedron but utilizes the more typical refine-then-project approach (see Fig. 2.14). Here, the employed projection is a specially designed projection that is also neither equal-area nor conformal—the ZOT (Zenithal Orthotriangular Projection (Dutton 1991))—which tries to have similar facets with vertices spaced uniformly in latitude and longitude, as well as low areal distortion. Indexing is performed similarly to the HSDS, with the faces of the initial octahedron indexed 1 through 8, and child cells indexed hierarchically in base 4.

SCENZ-Grid (SEEGrid 2019) is a DGG that is constructed based on an initial cube polyhedron, and which was created through a collaboration between Landcare Research and GNS Science primarily for the purpose of environmental monitoring.

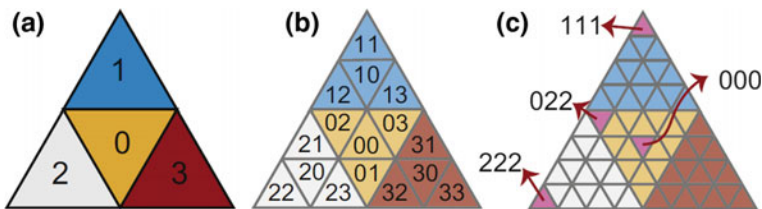


Fig. 2.13 (a, b) The hierarchical indexing system of the HSDS. (c) Indices of descendant cells after three refinements

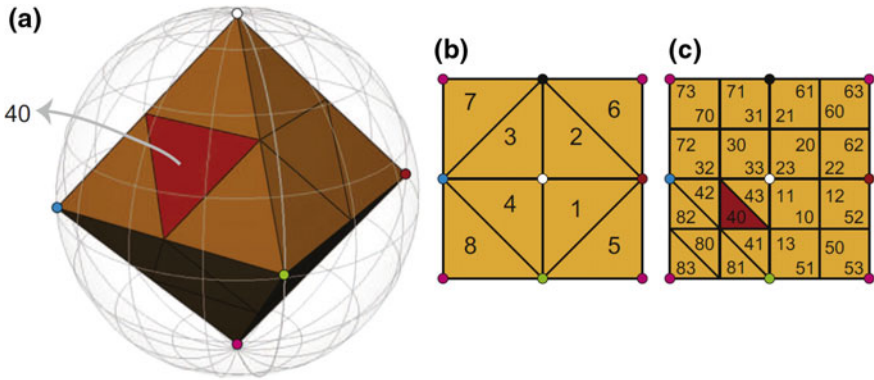


Fig. 2.14 The QTMDGGS. (a) The initial octahedron, embedded in a sphere. (b, c) The hierarchical indexing system of the QTM

The faces of the initial cube are refined using a 1-to-9 congruent and aligned quadrilateral refinement, and the resulting cells are inverse projected using the HEALPix projection method (see Sect. 2.2.4.2). A hierarchical base-9 indexing system is used to address the cells (see Fig. 2.15).

Quadrilateral cells are also found in Crusta (Bernardin et al. 2011), a DGGS based on a rhombic triacontahedron. Each of the initial 30 quadrilateral faces undergoes a 1-to-4 refinement, and the generated vertices are normalized to the geoid. Crusta’s primary motivation includes support for high-resolution topographical data and images.

A number of hexagon-based DGGSs have also been proposed and have garnered much research attention. The ISEA3H (Icosahedral Snyder Equal Area Aperture 3 Hexagonal) DGGS is a particularly notable example which starts from an icosahedron (or truncated icosahedron) that undergoes an aligned 1-to-3 hexagonal refinement (US Patent No. 8400451, 2004; Sahr 2008). The resulting cells are inverse projected to the sphere using Snyder’s equal-area projection. Note that as the refinement scheme

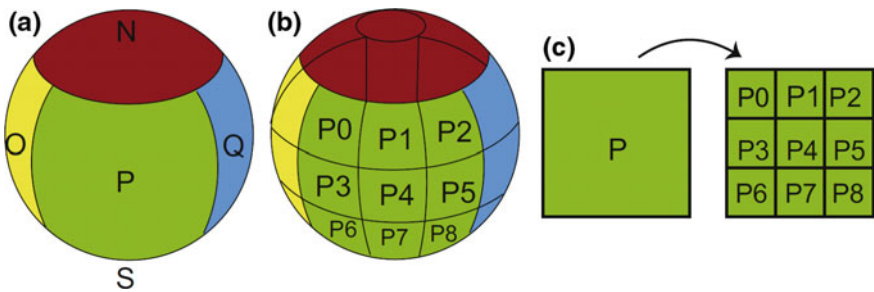
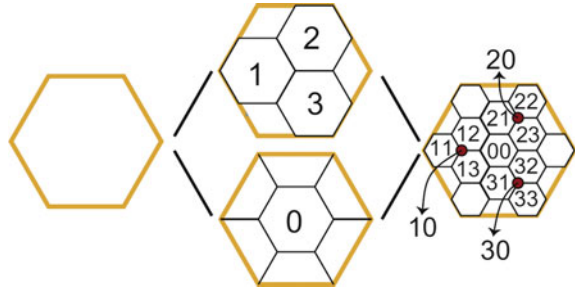


Fig. 2.15 (a), (b) SCENZ-Grid is created from a refined cube inverse projected to the sphere. (c) The hierarchical indexing system of SCENZ-Grid

Fig. 2.16 Two types of 1-to-4 hexagonal refinement can be combined, allowing a triangular cell hierarchy to be established and indexed



(and, indeed, any hexagonal refinement scheme) is not congruent, special care must be taken to define the cell hierarchy and indexing scheme. Several different indexing schemes have been proposed for the ISEA3H DGGs. These include the hierarchical indexing of PYXIS (US Patent No. 8400451, Peterson 2004), CPI (US Patent No. 9311350, Sahr 2016), coordinate-based indexing mechanisms (Sahr 2008; Mahdavi-Amiri et al. 2015a; Vince 2006), or the algebraic encoding scheme of (Ben et al. 2018).

The icosahedron can also be refined using 1-to-4 hexagonal refinement, as in the construction of the HQBS (Hexagonal Quaternary Balanced Structure (Tong et al. 2013)). The resulting cells are also inverse projected using Snyder’s projection. In order to mitigate the incongruity of the hexagonal refinement, two different 1-to-4 refinements are employed (aligned and unaligned; see Fig. 2.16). This allows a triangular hierarchy to be defined (aligned with the edges of the initial icosahedron) and for lattice points to be indexed. By taking the index of the point at the cell’s centroid to be the cell’s index, a base 4 hierarchical indexing system can be established on the cells.

Other hexagon-based DGGs include the OA3HDGG and OA4HDGG (Octahedral Aperture 3/4 Hexagonal Discrete Global Grid (Vince 2006; Ben et al. 2010)). As implied by the name(s), both DGGs are constructed from an octahedron that undergoes a hexagonal refinement. The OA3HDGG utilizes a 1-to-3 hexagonal refinement, and its cells are indexed using a coordinate-based system. The vertices of the initial octahedron are assigned the coordinates $(\pm 1, 0, 0)$, $(0, \pm 1, 0)$, and $(0, 0, \pm 1)$; and the cells are assigned indices based on their barycentric coordinates with respect to these vertices. A similar indexing system is applied to the OA4HDGG, which utilizes a 1-to-4 hexagonal refinement.

While most DGGs discretize only the surface of the Earth, certain types of geospatial data (e.g. earthquake data, airspace delineations, etc.) are volumetric in nature and require a volumetric Earth representation. Hence, volumetric DGGs such as SDOG (Spheroid Degenerated-Octree Grid (Yu and Wu 2009; Yu et al. 2012)) have also been proposed. SDOG, which was designed to represent the global lithosphere, divides the Earth into an initial set of eight octants. Each octant is associated with a degenerated octree, and undergoes a non-uniform refinement that prevents cell degeneracies at the Earth’s core (see Fig. 2.17). Cells are indexed using two different

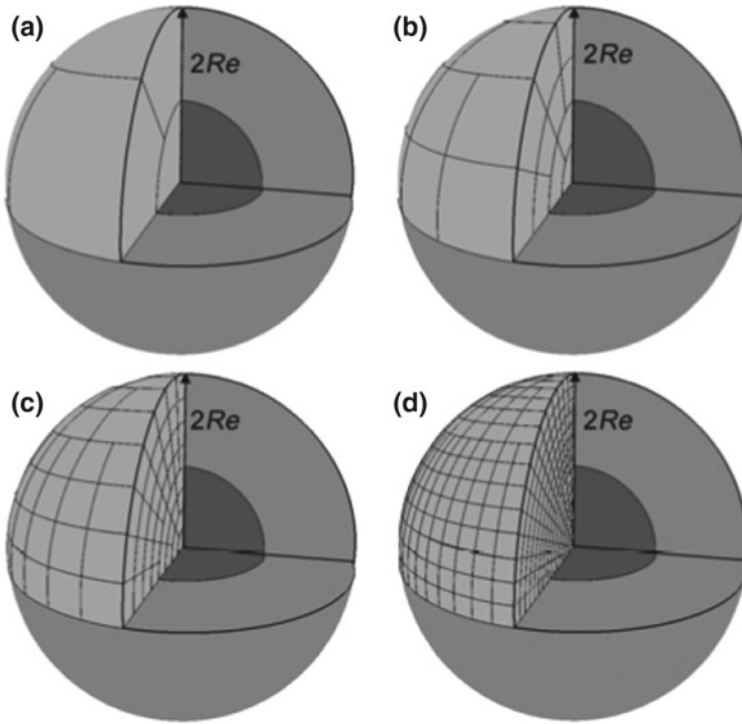


Fig. 2.17 The SDOG volumetric DGGS at three successive resolutions

curve-based schemes, both based on a modified Z-curve. The SDZ (Single Hierarchical Degenerated Z-Curve Filling) method indexes the cells of a single resolution in base 10 (see Fig. 2.18), while MDZ (Multiple Hierarchical Degenerated Z-Curve Filling) serves as a hierarchical indexing scheme in base 8 (see Fig. 2.19).

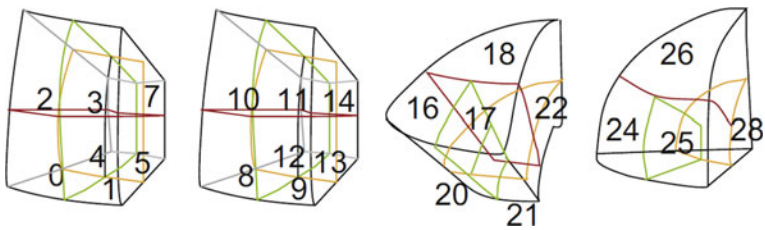


Fig. 2.18 SDZ defines a base 10 indexing for cells at a single resolution

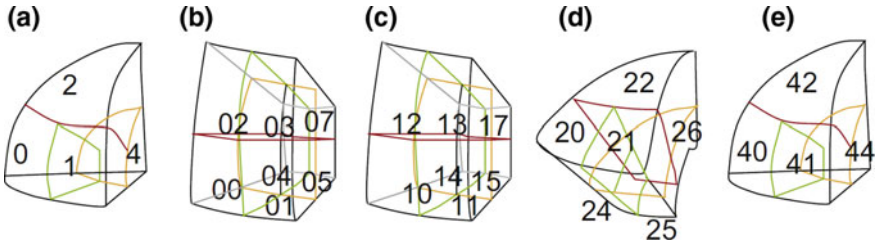


Fig. 2.19 MDZ defines a hierarchical indexing scheme on the SDOG cells. (a) A refined octant with child indices. (b), (c), (d) and (e) Cells 0, 1, 2, and 4, respectively, refined with child indices

2.4 Public and Commercial Digital Earth Platforms

Naturally, a number of DGGs and other Digital Earth concepts have been implemented and made available for public use as either free or paid software.

2.4.1 Latitude/Longitude Grids

Due to their ease of use and long history, latitude/longitude grids remain a popular choice for Digital Earth implementations despite the potential issues associated with non-uniform DGGs cells. Chief among these implementations in terms of name recognition is Google Earth (Google Inc. 2019a). Google Earth is created upon a latitude/longitude grid using a simple cylindrical projection, with textures processed via clip-mapping (Bar-Zeev 2007). Clip-Maps are a modified form of mip-map that impose a maximum image size on the mip-map hierarchy (Tanner et al. 1998), causing the image hierarchy to more closely resemble an obelisk than the traditional pyramid (see Fig. 2.20). The capped image size ensures that textures can fit into memory and be rendered in real-time.

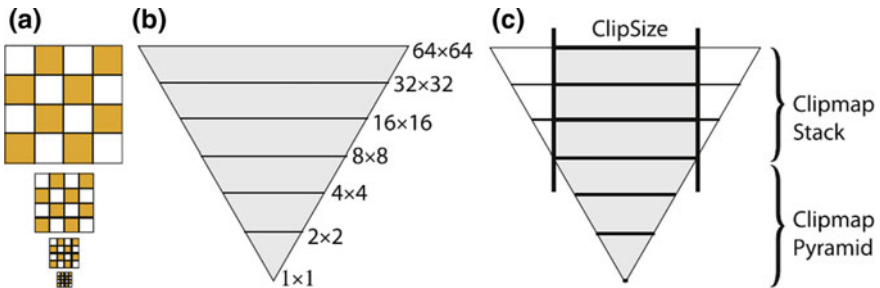


Fig. 2.20 (a) An image at multiple resolutions. (b) The image's mip-map pyramid. c Clip-Maps impose a maximum size on the mip-maps

Although not presented in 3D globe format, Google Maps and Bing Maps are supported using methods that echo the fundamentals of a DGGS (See Figs. 2.21 and 2.22). In particular, Google Maps uses the Mercator projection on a latitude/longitude grid that is refined using a 1-to-4 refinement. Each “tile” of the hierarchical grid is associated with a 256×256 -sized texture, and is indexed using an axis-based coordinate system. Here, the top-left tile is indexed as $(0, 0)$, with x values increasing towards the east, and y values increasing towards the south (Google Inc. 2019b).

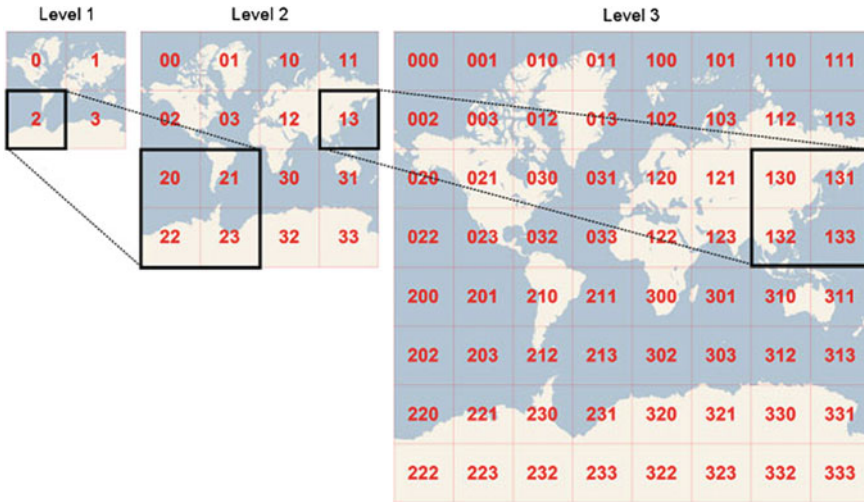


Fig. 2.21 Three resolutions of Bing Maps’ hierarchy-based indexing

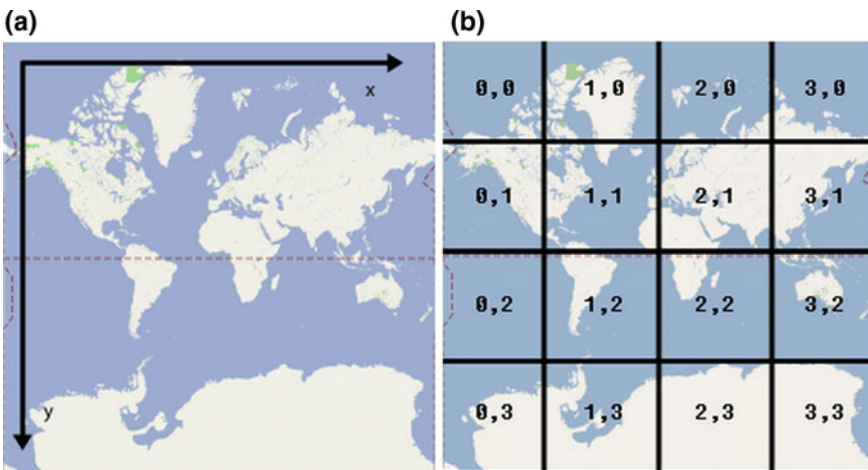


Fig. 2.22 (a) Google Maps’ latitude/longitude grid. (b) Cell indices at the second resolution

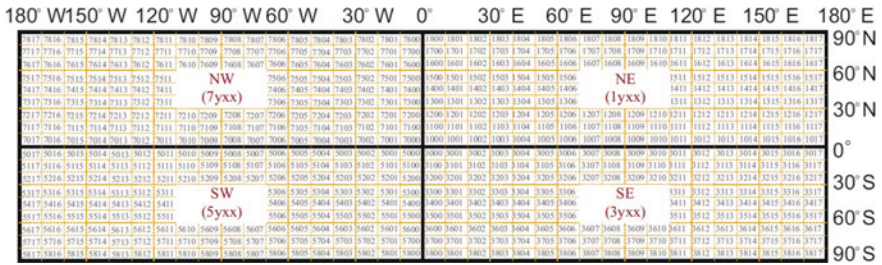


Fig. 2.23 C-squares indexing system

Bing Maps also uses the Mercator projection on a 1-to-4 refined grid, but its indexing system is hierarchical and based on quadtrees (Schwartz 2018). For an illustration, see Fig. 2.21.

The OGC CDB (Common Database) API from Presagis (2019) is designed to address one of the main issues with latitude/longitude DGGs, namely the shrinking of cells near the poles. The CDB divides the Earth into five zones depending on proximity to the poles, with each zone utilizing a different spacing between lines of longitude. While the CDB is available as an open commercial standard, a Pro license can be purchased for additional features.

Unlike other DGGs, C-squares (Concise Spatial Query and Representation System (Rees 2003)) discretizes only a single resolution of the Earth. Here, the latitude/longitude grid is divided into four quadrants (NE, NW, SE, and SW), which are then divided into finer grids based on latitude and longitude (Fig. 2.23). The cells of this discretization are indexed as *ixxx*, where *i* corresponds to the cell’s quadrant, *y* to the cell’s latitude, and *xx* to the cell’s longitude. This system was created by CSIRO Marine and Atmospheric Research, Australia for the purposes of mapping, spatial search, and environmental assessment. Converters and source code can be found on their website (CSIRO 2019).

Other Digital Earths based upon latitude/longitude grids include NASA’s open source WorldWind API (NASA 2019); Skyline’s software products, TerraExplorer client and SkylineGlobe server (Skyline Software Systems 2019); and two DGGs libraries for web-based globe visualization—GlobWeb and CesiumJS (Telespazio 2019; Cesium Consortium 2019). GlobWeb is provided by Telespazio France under the GNU LGPL v3 license, while CesiumJS was funded by the Cesium Consortium and is open source.

CesiumJS in particular is a complete 3D mapping platform built using WebGL. It is a cross-platform and cross-browser map engine that runs on a web browser without plugins, and is now used in industries as diverse as archaeology, engineering, construction, and sports visualizations. An accompanying tool, Cesium ion, provides a point-and-click workflow to create 3D maps of users’ geospatial data that can be visualized, analysed, and shared. It can be used to host datasets in 3D tiles, including imagery, terrain, photogrammetry, point clouds, BIM, CAD, 3D buildings, and vector

data; and provides tools for analytics including measurements, volume and visibility computations, and terrain profiles.

2.4.2 Geodesic DGGs

Of course, not all DGGs are based on singular 2D domains such as a latitude/longitude grid; while comparatively rarer, different implementations of Geodesic DGGs do exist and are available for use. For instance, a library that implements the well-studied ISEA3H DGGs—known as *geogrid*—is offered on GitHub (Mocnik 2019). This library is developed and maintained by Franz-Benjamin Mocnik, and is licensed under the MIT license.

A propriety implementation of the ISEA3H can be found at the core of the Digital Earth system developed by Global Grid Systems (formerly the PYXIS innovation (PYXIS innovation 2011; Global Grid Systems 2019)), and is one of the few commercially available Geodesic DGGs (Fig. 2.24). This system indexes the ISEA3H’s hexagonal cells using the patented PYXIS indexing scheme (US Patent No. 8400451, 2004).

Other software platforms include implementations of the ECM (Ellipsoidal Cube Map) and HEALPix (Hierarchical Equal Area isoLatitude Pixelization of the sphere) DGGs. ECM (Lambers and Kolb 2012) is produced by applying 1-to-4 refinement on the quadrilateral faces of a cube that circumscribes the ellipsoidal Earth. Areal and angle distortions are minimized by using a Quadrilateralized Spherical Cube (QSC) projection. A Linux implementation is available on Martin Lambers’ website, licensed under the GNU GPL v3 (Lambers 2019).

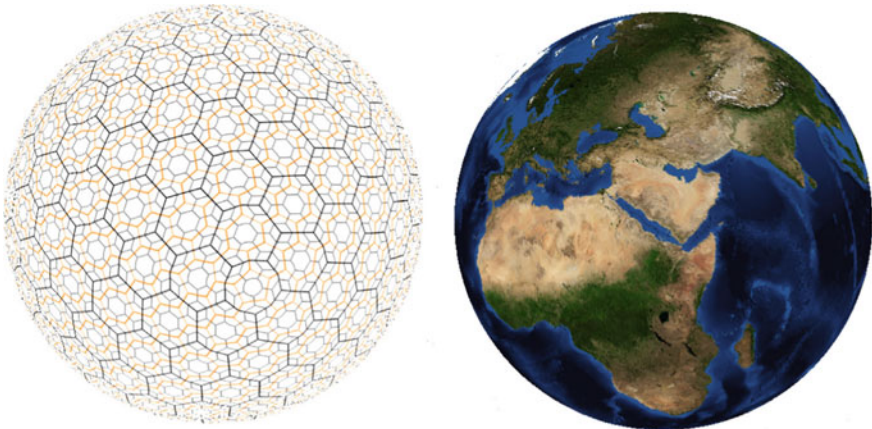


Fig. 2.24 Global Grid Systems’ ISEA3H DGGs

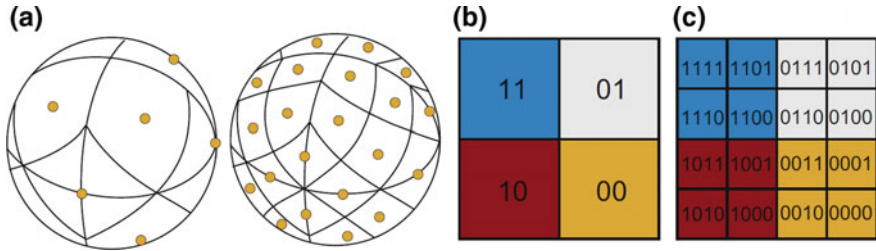


Fig. 2.25 The HEALPix DGGs. (a) HEALPix uses a 1-to-4 refinement. (b) The hierarchical HEALPix indexing system

The HEALPix DGGs (Gorski et al. 2005) is based on a rhombic dodecahedron that undergoes a congruent 1-to-4 refinement, indexed using a base-2 hierarchical indexing system (see Fig. 2.25). Two different projection schemes are employed: Lambert cylindrical equal-area projection for equatorial regions, and Collignon equal-area projection for polar regions. A software package from the Jet Propulsion Laboratory that supports spherical harmonics, pixel queries, data processing, and statistical analysis can be found online (Jet Propulsion Laboratory 2019).

2.4.3 Installations: DESP

One of the largest scale Digital Earth undertakings can be found at the Chinese Academy of Sciences (CAS), where an interactive visualization environment called the Digital Earth Science Platform was developed (Guo et al. 2017). Based on the Digital Earth Prototype System Initiative that launched in 1999 (Guo et al. 2009, 2010), the Digital Earth Science Platform (DESP) was established by the CAS in 2010 in order to integrate state-of-the-art techniques and meet the increasing requirements of geoscience applications.

The DESP is a technical platform that supports spatial data and information services, as well as associated applications. It integrates 2D and 3D geographic information systems, distributed storage and computing, virtual reality, wireless sensor networks, and other technologies. A 600 m² fully immersive, interactive visualization environment was established at the CAS Institute of Remote Sensing and Digital Earth (RADI) to support experiments with 3D visualization and to provide decision support for emergency response applications. This installation is equipped with VR/AR devices, sensors, a 3D Stereo Projection System, and a high-performance computing system, as shown in Fig. 2.26.

The DESP has already played an important role in the modeling of global change, evaluation of natural disasters, and monitoring of natural resources and human settlement through the integration of multi-sensor, multi-temporal remote sensing images, in situ ground survey data, socio-economic data, and interdisciplinary scientific models (Fan et al. 2009). For example, the influence of sea level rise on the Earth’s major



Fig. 2.26 The DESP visualization environment at RADI

river deltas has been modeled and analyzed in a comparative study by using the DESP. Emergency monitoring and response systems based upon the DESP have been developed for disaster monitoring and post-disaster relief after earthquake events (Fig. 2.27).

As a part of the ongoing Big Earth Data Science Engineering (CASEarth) initiative (2018–2022), which is supported by the Strategic Priority Research Program of the CAS, a new generation of the Digital Earth Science Platform will be developed to provide a new impetus for interdisciplinary, cross-scale, macro-scientific discoveries in the Era of Big Data to promote sustainability (Guo 2017).



Fig. 2.27 The DESP was used for disaster assessment and relief after the 2013 Ya’an earthquake

2.5 Discrete Global Grid System Standards

The myriad ways in which one can construct a Digital Earth platform provide a great deal of flexibility that can help cater to a vast range of specific uses; however, this can also create barriers to interoperability. This creates a real challenge as we move into the Era of Big Data (and beyond), where interoperability and distributed analysis is critical.

In the Era of Big Data, geoscience can only achieve its full potential through the fusion of diverse Earth observation and socio-economic data together with information from a vast range of sources. In this type of environment with multiple data providers, fusion is only possible with an information system architecture based upon open standards (Percivall 2013). Without a common and standardized means of defining and integrating various Digital Earth Platforms, our ability to transform the increasingly massive amounts of data that are being acquired about the Earth into actionable information is significantly limited.

2.5.1 *Standardization of Discrete Global Grid Systems*

Recognizing the issues that non-standard global grid system implementations pose and their potential impacts, in 2014, the Open Geospatial Consortium embarked on the ambitious goal of standardizing DGGS. The goal of this endeavor was not to identify the one DGGS that ought to be used by everyone, but to define the common qualities of a variety of DGGSs that can be used to support interoperability while providing some flexibility in choice, thus allowing implementers to tailor DGGS infrastructures to their specific uses. In July 2017, the OGC published the first ever international standard governing the design and implementation of DGGS (Purs et al. 2017). This standard aims to promote awareness and reusability of DGGS implementations, and integration between them, and, through this, to demonstrate a path towards the realization of the “Digital Twin”—where our engagement and understanding of the physical Earth can seamlessly interact with the Digital Earth, and vice versa.

The core of the OGC DGGS standard is primarily based on an appropriate subset of the well-accepted criteria for optimal DGGS design proposed by Goodchild (2000) and Sahr et al. (2003).

2.5.2 *Core Requirements of the OGC DGGS Abstract Specification*

Along with the categorization provided earlier, under the OGC DGGS Abstract Specification, a compliant DGGS must define a hierarchical tessellation of equal area

cells that both partition the entire Earth at multiple levels of granularity and provide a global spatial reference frame. In addition to these structural components, the system must also include encoding methods to address each cell, assign quantized data to cells, and perform algebraic operations on the cells and the data assigned to them.

The requirement of functional components for the infrastructure sets an OGC DGGS apart from other grid frameworks or Coordinate Reference Systems. It also provides a common operational basis for supporting communication and interoperability between different compliant DGGS infrastructures.

2.5.2.1 Structural Requirements

The reference frame of a DGGS consists of the fixed structural elements that define the spatial framework on which the DGGS functional algorithms operate. These fixed structural elements include:

1. **Domain completeness and position uniqueness:** The DGGS must be defined over a global domain without any overlapping cells. Goodchild defines a global domain to be achieved when the areal cells defined by the grid “*exhaustively cover the globe without overlapping or underlapping*” (Goodchild 2000);
2. **Multiple levels of resolution:** The DGGS must define multiple discrete global grids forming a system of hierarchical tessellations, each with progressively finer spatial resolution and linked via a common cell refinement method;
3. **Preservation of domain completeness and position uniqueness:** The DGGS must preserve the total surface area (i.e. the global domain) throughout the entire range of hierarchical tessellations. This facilitates the consistent representation of information at all resolutions within the DGGS;
4. **A simple geometric structure for each cell:** In order for the DGGS to achieve the requirement of a global domain, it is necessary for the shape of all cells defined by the DGGS to be simple polygons on the surface model of the Earth. The cell shapes derived from the five (5) Platonic solids and thirteen (13) Archimedean solids (triangle, quadrilateral, pentagon, hexagon, and octagon) are all simple polygons that have the following properties:
 - a. The edges meet only at the vertices;
 - b. Exactly two edges meet at each vertex; and,
 - c. The polygons enclose a region which always has a measurable area.
5. **Equal-area cells:** The DGGS must be based on a hierarchy of equal-area tessellations. Equal-area cells provide global grids with spatial units that (at multiple resolutions) have an equal probability of contributing to an analysis. Equal-area cells also help minimize the confounding effects of area variations in spatial analyses, where the curved surface of the Earth is the fundamental reference frame;
6. **An initial polyhedral tessellation:** To consistently achieve equal-area cells, the DGGS must be constructed by mapping a polyhedron to the surface model of

the Earth. This initial tessellation can then be refined to produce equal-area child cells for all subsequent levels in the hierarchy of tessellations;

7. **Unique identifiers for each cell:** In order to efficiently operate as a spatial data integration engine, the cells of the DGGS must each be defined by a globally unique identifier. This ensures that the reference to each and every cell is immutable. While the OGC DGGS Abstract Specification requires each cell to be uniquely identified, it does not prescribe or enforce how the implementer must achieve this;
8. **Each cell referenced at its centroid location:** Each DGGS cell must be referenced at its centroid. This is because the centroid is the only location that provides a systematic and consistent spatial reference point for all cells, regardless of shape.

2.5.2.2 Functional Requirements

The ability to locate and perform algebraic operations on data assigned to a DGGS is critical for a DGGS to be able to support connectivity and hierarchical operations on cells and to facilitate interoperability between DGGS implementations (as well as other spatial data infrastructures or interfaces). Accordingly, the OGC DGGS Abstract Specification requires the DGGS to specify definitions for:

1. **Quantization operations:** Assigning data to and retrieving data from cells;
2. **Algebraic operations:** Performing algebraic operations on cells and the data assigned to them, in addition to performing cell navigation; and
3. **Interoperability operations:** Translating cell addresses to other Coordinate Reference Systems (CRS), such as conventional latitude/longitude.

Again, the OGC DGGS Abstract Specification enforces no specific implementations of these functional elements, but requires their inclusion (in some form) in any compliant DGGS implementation. This both facilitates flexibility and innovation in the design of individual DGGS implementations and ensures the widest scope for interoperability between compliant DGGS implementations. By focusing on endpoint functional requirements and not on the methods by which they are achieved, the OGC DGGS Abstract Specification supports interoperability across multiple social and technical domains. This approach also allows for advancements in the technologies that support these functional elements, without requiring the standard to be constantly re-written.

2.5.3 *The Future of the DGGS Standard*

In support of the wider adoption and implementation of compliant (and interoperable) DGGS implementations, there are a number of initiatives currently underway within

both the OGC and the International Standards Organization (ISO). These initiatives include:

1. The publication of the OGC DGGs Abstract Specification as an ISO standard (ISO 19170). By publishing this standard as an ISO standard, it will be possible to reach a wider community of potential DGGs implementers and thus increase the adoption of DGGs technologies.
2. The establishment of an OGC Registry of compliant DGGs implementations. This will facilitate the certification and publication of compliant DGGs implementations and increase the awareness of the choices of available DGGs implementations that can be applied to a Spatial Data Infrastructure. This will be similar in nature to the Coordinate Reference System Registry. The first release of the OGC DGGs Registry is anticipated to occur by the end of 2018.
3. The development of a standardized specification of a common API language for DGGs. This work is in its early phase but is expected to result in the drafting and publication of a new OGC implementation standard that specifies a common API language supporting and facilitating interoperability between different DGGs implementations. A common API language for DGGs implementations will further lower the technical barriers to the wider implementation of DGGs technologies.

2.5.4 Linkages Between DGGs and Other Standards Activities

As a technology, DGGs have the potential to impact on almost all spatial technologies and their related standards. Consequently, a number of international standards activities have included references to DGGs and their potential applications to several scenarios relevant to these initiatives. Two examples of this include:

1. The Joint OGC-W3C Spatial Data on the Web Best Practices (Van Den Brink et al. 2019), where DGGs was proposed as an enabling component of QB4ST (an extension of existing RDF Datacube vocabularies to support spatio-temporal data).
2. The Global Statistical Geospatial Framework (GSGF), adopted during the 6th Session of the United Nations Committee of Experts on Global Geospatial Information Management in August 2016, refers to DGGs and acknowledges that these technologies have the potential to help realize the implementation of the GSGF.

As the number of DGGs implementations increases, so too will the suite of international standards that support them and their applications. The challenge for the International Standards Community will be to keep the number and complexity of these standards to an acceptable level in order to ensure that the DGGs standards do not become a barrier to adoption in themselves.

Acknowledgements Elements of this chapter are inspired by or reproduced from:

- Mahdavi-Amiri, A., Alderson, T., & Samavati, F (2015). A Survey of Digital Earth. *Computers and Graphics, 53 (Part B)*, 95–117.
- Foster, C., Purss, M., Peterson, P., Gibb, R., Oliver, S., Samavati, F., Woodcock, R., Evans, B., OGC Discrete Global Grid System (DGGS) Core Standard, 2015.
- Mahdavi-Amiri, A., ACM: Atlas of Connectivity Maps. Ph.D. Thesis, Department of Computer Science, University of Calgary, 2015.

References

- Bai J, Zhao X, Chen J (2005) Indexing of the discrete global grid using linear quadtree. In: Jiang J (ed) ISPRS workshop on service and application of spatial data infrastructure, Hangzhou, 14–16 October 2005. ISPRS, pp 267–270
- Bartholdi JJ, Goldsman P (2001) Continuous indexing of hierarchical subdivisions of the globe. *Int J Geogr Inf Sci* 15(6):489–522
- Bar-Zeev A (2007) How Google Earth [really] works. Reality Prime. <http://www.realityprime.com/blog/2007/07/how-google-earth-really-works>. Accessed 30 July 2019.
- Ben J, Tong X, Chen R (2010) A spatial indexing method for the hexagon discrete global grid system. In: 18th international conference on geoinformatics, IEEE, Beijing, 18–20 June 2010
- Ben J, Li Y, Zhou C et al (2018) Algebraic encoding scheme for aperture 3 hexagonal discrete global grid system. *Sci China Earth Sci* 61(2):215–227
- Bernardin T, Cowgill E, Kreylos O et al (2011) Crusta: a new virtual globe for real-time visualization of sub-meter digital topography at planetary scales. *Comput Geosci* 37(1):75–85
- Cesium Consortium (2019) CesiumJS - Geospatial 3D mapping and virtual globe platform. <https://cesiumjs.org>. Accessed 30 July 2019
- Cozzi P, Ring K (2011) 3D engine design for virtual globes. CRC Press, Hoboken
- CSIRO (2019) C-squares home page. <http://www.cmar.csiro.au/csquares>. Accessed 30 July 2019
- Dutton G (1991) Zenithial orthotriangular projection: a useful if unesthetic polyhedral map projection to a peculiar plane. In: 10th international symposium on computer-assisted cartography. Auto-Carto 1991, Baltimore, 25–28 March 1991. ACSM and ASPRS, Bethesda, pp 77–95
- Dutton GH (1999) A hierarchical coordinate system for geoprocessing and cartography. Springer, New York
- Fan X, Du X, Tan J et al (2009) Three-dimensional visualization simulation assessment system based on multi-source data fusion for the Wenchuan earthquake. *J Appl Remote Sens* 3(1):1–9
- Fekete G, Treinish LA (1990) Sphere quadtrees: a new data structure to support the visualization of spherically distributed data. In: Farrell EJ (ed) Proceedings volume 1259, extracting meaning from complex data: processing, display, interaction, Santa Clara, 11–16 February 1990. SPIE, pp 242–254
- Gargantini I (1982) An effective way to represent quadtrees. *Commun ACM* 25(12):905–910
- Global Grid Systems (2019) Global grid systems. <https://www.globalgridsystems.com>. Accessed 30 July 2019
- Goodchild MF (2000) Discrete global grids for digital Earth. In: International conference on discrete global grids, NCGIA, Santa Barbara, 26–28 March 2000
- Goodchild MF, Shiren Y (1992) A hierarchical spatial data structure for global geographic information systems. *CVGIP: Gr Model Image Process* 54(1):31–44
- Google Inc (2019a) Google Earth. <https://www.google.com/earth>. Accessed 30 July 2019
- Google Inc (2019b) Tile overlays | maps SDK for android | Google developers. <https://developers.google.com/maps/documentation/android-sdk/tileoverlay>. Accessed 30 July 2019
- Gorski KM, Hivon E, Banday AJ et al (2005) HEALPix: a framework for high-resolution discretization and fast analysis of data distributed on the sphere. *Astrophys J* 622(2):759–771

- Grafarend EW, You RJ, Syffus R (2014) Map projections: cartographic information systems. Springer, Heidelberg
- Guo H (2017) Big Earth data: a new frontier in Earth and information sciences. *Big Earth Data* 1(1–2):4–20
- Guo H, Fan X, Wang C (2009) A digital Earth prototype system: DEPS/CAS. *Int J Digit Earth* 2(1):3–15
- Guo HD, Liu Z, Zhu LW (2010) Digital Earth: decadal experiences and some thoughts. *Int J Digit Earth* 3(1):31–46
- Guo H, Liu Z, Jiang H et al (2017) Big Earth Data: a new challenge and opportunity for Digital Earth's development. *Int J Digit Earth* 10(1): 1–12
- Harrison E, Mahdavi-Amiri A, Samavati F (2011) Optimization of inverse snyder polyhedral projection. In: 2011 international conference on cyberworlds, IEEE, Banff, 4–6 October 2011
- Harrison E, Mahdavi-Amiri A, Samavati F (2012) Analysis of inverse snyder optimizations. In: Gavrilova ML, Tan CJK (eds) *Transactions on computational science XVI*. Springer, Heidelberg, pp 134–148
- Jet Propulsion Laboratory (2019) Jet propulsion laboratory HEALPix home page. <https://healpix.jpl.nasa.gov>. Accessed 30 July 2019
- Lambers M (2019) ECM: ellipsoidal cube maps. <https://marlam.de/ecm>. Accessed 30 July 2019
- Lambers M, Kolb A (2012) Ellipsoidal cube maps for accurate rendering of planetary-scale terrain data. In: Bregler C, Sander P, Wimmer M (eds) *20th Pacific conference on computer graphics and applications (short papers)*. Pacific Graphics 2012, Hong Kong, 12–14 September 2012. The Eurographics Association, pp 5–10
- Mahdavi-Amiri A, Samavati F (2012) Connectivity maps for subdivision surfaces. In: Richard P, Kraus M, Laramée R, Braz J (eds) *International conference on computer graphics theory and applications. GRAPP 2012*, Rome, 24–26 February 2012. SciTePress, pp 26–37
- Mahdavi-Amiri A, Samavati F (2014) Atlas of connectivity maps. *Comput Gr* 39:1–11
- Mahdavi-Amiri A, Bhojani F, Samavati F (2013) One-to-two digital Earth. In: Bebis G, Boyle R, Parvin B et al (eds) *International symposium on visual computing. SVC 2013: advances in visual computing*. Springer, Heidelberg, pp 681–692
- Mahdavi-Amiri A, Harrison E, Samavati F (2015a) Hexagonal connectivity maps for digital Earth. *Int J Digit Earth* 8(9):750–769
- Mahdavi-Amiri A, Samavati F, Peterson P (2015b) Categorization and conversions for indexing methods of discrete global grid systems. *ISPRS Int J Geo-Inf* 4(1):320–336
- Middleton L, Sivaswamy J (2005) *Hexagonal image processing: a practical approach*. Springer, London
- Mocnik F-B (2019) GitHub - GIScience/geogrid: library for discrete global grid systems. <https://github.com/GIScience/geogrid>. Accessed 30 July 2019
- NASA (2019) NASA WorldWind. <https://worldwind.arc.nasa.gov/>. Accessed 30 July 2019
- Percival G (2013) Geodata fusion study by the open geospatial consortium. In: Pellechia MF, Sorensen RJ, Palaniappan K (eds) *Proceedings volume 8747, geospatial InfoFusion III*, Baltimore, 29 April–3 May 2013. SPIE, pp 87470A-1–87470A-13
- Peterson P (2004) Close-packed, uniformly adjacent, multiresolutional, overlapping spatial data ordering. US Patent 8400451, 19 March 2013
- Presagis (2019) OGC CDB - Presagis. <https://www.presagis.com/en/glossary/detail/ogc-cdb>. Accessed 30 July 2019
- Purss M, Gibb R, Samavati F et al (2017) Discrete global grid systems abstract specification – Topic 21. In: Purss M (ed). *Open geospatial consortium*, Wayland
- Purss M, Peterson P, Strobl P et al (2019) Datacubes: a discrete global grid systems perspective. *Cartographica* 54(1):63–71
- PYXIS Innovation (2011) How PYXIS works - Pyxis public wiki. http://www.pyxisinnovation.com/pyxwiki/index.php?title=How_PYXIS_Works. Accessed 30 July 2019
- Rees T (2003) “C-squares”, a new spatial indexing system and its applicability to the description of oceanographic datasets. *Oceanography* 16(1):11–19

- Roşca D, Plonka G (2011) Uniform spherical grids via equal area projection from the cube to the sphere. *J Comput Appl Math* 236(6):1033–1041
- Roşca D, Plonka G (2012) An area preserving projection from the regular octahedron to the sphere. *Results Math* 62(3):429–444
- Sahr K (2008) Location coding on icosahedral aperture 3 hexagon discrete global grids. *Comput Environ Urban Syst* 32(3):174–187
- Sahr K (2016) Central place indexing systems. US Patent 9311350, 12 April 2016
- Sahr K, White D, Kimerling AJ (2003) Geodesic discrete global grid systems. *Cartogr Geogr Inf Sci* 30(2):121–134
- Schwartz J (2018) Bing maps tile system. Microsoft Docs. <https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system>. Accessed 30 July 2019
- SEEGrid (2019) WebHome<SCENZGrid<SEEGrid. <https://www.seegrid.csiro.au/wiki/SCENZGrid/WebHome>. Accessed 30 July 2019
- Skyline Software Systems (2019) Skyline software systems: skylineglobe 3D earth software solutions. <http://www.skylineglobe.com>. Accessed 30 July 2019
- Snyder J (1992) An equal-area map projection for polyhedral globes. *Cartographica* 29(1):10–21
- Tanner C, Migdal C, Jones M (1998) The clipmap: a virtual mipmap. In: Cunningham S, Bransford W, Cohen MF et al. (eds) *Proceedings of the 25th annual conference on computer graphics and interactive techniques*. ACM, Orlando, pp 19–24
- Telespazio (2019) GlobWeb. <https://github.com/TPZF/GlobWeb>. Accessed 30 July 2019
- Tobler W, Chen Z-T (1986) A quadtree for global information storage. *Geogr Anal* 18(4):360–371
- Tong X, Ben J, Wang Y et al (2013) Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system. *Int J Geogr Inf Sci* 27(5):898–921
- van den Brink L, Barnaghi P, Tandy J et al (2019) Best practices for publishing, retrieving, and using spatial data on the web. *Semant Web* 10(1):99–114
- Vince A (2006) Indexing the aperture 3 hexagonal discrete global grid. *J Vis Commun Image Represent* 17(6):1227–1236
- Vince A, Zheng X (2009) Arithmetic and Fourier transform for the PYXIS multi-resolution digital Earth model. *Int J Digit Earth* 2(1):59–79
- White D (2000) Global grids from recursive diamond subdivisions of the surface of an octahedron or icosahedron. *Environ Monit Assess* 64(1):93–103
- White D, Kimerling JA, Overton SW (1992) Cartographic and geometric components of a global sampling design for environmental monitoring. *Cartogr Geogr Inf Syst* 19(1):5–22
- Yu J-Q, Wu L-X (2009) Spatial subdivision and coding of a global three-dimensional grid: Spheroid degenerated-octree grid. In: 2009 IEEE international geoscience and remote sensing symposium, IEEE, Cape Town, 12–17 July 2009
- Yu JQ, Wu LX, Zi GJ et al (2012) SDOG-based multi-scale 3D modeling and visualization on global lithosphere. *Sci China Earth Sci* 55(6):1012–1020

Troy Alderson is a Ph.D. candidate in the Department of Computer Science at the University of Calgary, where he also received his M.Sc. in Computer Science (2014) and B.Sc.'s in Computer Science and General Mathematics (2010). His current work focuses on developing unified frameworks for subdivision and multiresolution for various types of geometric shapes.

Matthew Purss is a founding co-chair of the Open Geospatial Consortium's Discrete Global Grid Systems working groups. He holds a Ph.D. in Exploration Geophysics and throughout his 20+ year career he has been a leader in the theoretical development, practical implementation and standardisation of advanced spatial data infrastructures and grid systems.

Xiaoping Du is currently an Associate Professor with the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences. His research interests include Digital Earth and geospatial big data analytics, coastal flooding risk assessment due to global climate change, and remote sensing for natural disasters & environmental monitoring.

Ali Mahdavi-Amiri is a University Research Associate at Computing Science department in Simon Fraser University. His research interests is in Computer Graphics with specialty in Geometric Modeling, Geospatial Modeling and Visualization, Digital Fabrication, and Machine Learning. Ali has received several awards including NSERC Postdoctoral Fellowship, Graduate Achievement Award, Michael A. J. Sweeney Award, and Teaching Excellence Award.

Faramarz Samavati is a professor of Computer Science at the University of Calgary. Dr. Samavati's research interests include Computer Graphics, Visualization, and Digital Earth. Over the past eight years, he has received seven best paper awards, Digital Alberta Award, Great Supervisor Award, and University of Calgary Peak Award.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

