A Survey of Digital Earth

Ali Mahdavi-Amiri[a], Troy Alderson[a], Faramarz Samavati[a]

[a] *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 the GIS, OGC, and Digital Earth communities but have not been well-introduced to the computer graphics community. In this paper, we provide an overview of DGGSs and their use in digitally representing the Earth, describe several current Digital Earth systems and their methods of Earth representation, and list a number of applications of Digital Earths with related works. Moreover, we discuss the key research areas and related papers from computer graphics that are useful for a Digital Earth system, such as advanced techniques for geospatial data creation and representation.

1. Introduction

Over the past three decades, the field of computer graphics has experienced significant research and technical advancements, which have revolutionized application areas such as the film and video game industries. Inspired by this success, Thomas Funkhouser, in his visionary talk at the SIGGRAPH 2014 achievement award session, encouraged computer graphics experts to expand the scope of this field to new real-life application areas [1]. We focus on one such application, the Digital Earth, which is a disruptive approach to the methods of geospatial analysis and visualization currently employed within the field of GIS that uses a virtual (3D) representation of the globe as a reference model for geospatial data.

The Earth is immense, and data about the Earth is similarly immense. The field of GIS exists to gather, integrate, process, visualize 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 that 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, integrate, and distribute the data, is unsustainable in the face of the ever increasing flow of data.

Furthermore, even after the data are integrated, their traditional representation as layers of 2D maps presents difficulties in analyzing and understanding these data. While cartography is an old and studied field, human intuitions on distance and area break down in the face of areal distortions inherent in any projection of the Earth to a 2D cartographic map, especially for non-experts. Such maps offer a flat view of the 3D globe and do not take advantage of the possibilities for the interactive 3D exploration of the Earth that may be easily accomplished with modern graphics applications.

Moving to a 3D representation of the Earth for integration, analysis, and visualization can help address these issues, despite the fact that the non-Euclidean nature of the Earth's spherical surface presents its own challenges. One such representation that has emerged to facilitate solutions to data integration and analysis is that of the Digital Earth (aka the Digital or Virtual Globe). In a Digital Earth system, data are assigned to

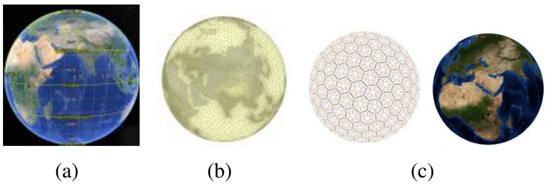


Figure 1: The Earth can be discretized into cells using, e.g., (a) latitude/longitude parametrization; Image taken from Google Earth [5], (b) Voronoi cells; Image taken from [3], or (c) a refined polyhedron projected to the sphere; Image taken from PYXIS Innovation Inc. framework [6].

locations on the 3D Earth (typically approximated using either a sphere or, for more accuracy, an ellipsoid) through one of a variety of techniques.

In most cases, data are assigned to the cells of an underlying discretization of the Earth (see Figure 1). Each cell represents a particular region and receives a unique index (or address), which can be used for fast data access and/or hierarchical or adjacency queries. While alternatives to discretization are possible, grappling with the non-Euclidean geometry of the Earth's surface is a non-trivial task. Examples of discretization approaches include Voronoi cells or latitude/longitude grids [2, 3, 4].

One of the advantages of latitude/longitude cells over Voronoi cells is that the cells are mostly *regular* (i.e. they have a grid-like structure), simplifying adjacency and hierarchical queries substantially. Discretizations of the Earth into a multiresolution hierarchy of indexed (mostly) regular cells are known as Discrete Global Grid Systems (DGGSs) and form the backbone of the state-of-the-art in Digital Earth systems. To further simplify such queries, most DGGSs aim for uniform, equal-area cells and are typically constructed via an initial discretization into planar cells that are then refined by some refinement method and projected onto the spherical Earth.

DGGSs may be characterized according to the shapes of the cells and their initial structure, the refinements applied to these cells, the projection method that maps points on a planar map to and from 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 the state-of-the-art Digital Earth software currently in use and applications of globe representation and visualization, for use by both the computer graphics and GIS communities.

The survey is organized as follows. In Section 2, we review the data types typically encountered in a Digital Earth setting, alongside relevant research and survey papers that deal with using, managing, and creating such data sets. The various types of DGGSs and their components are discussed in Section 3, followed by an overview of existing and well-known Digital Earth systems and each's underlying globe representation in Section 4. Finally, we mention some of the important applications of the Digital Earth in the modern age in Section 5 and conclude in Section 6.

2. Data Types

Various types of geospatial data sets are visualized, analyzed, and combined on a Digital Earth. Although these data sets generally fall into a small number of type categories, there exist a plethora of file formats used to describe and store them (see [7]), data acquisition techniques used to generate them, and organizations that collect and catalog them. Each category of data type is a field of research unto itself and each has been studied in several works within the literature.

One of the main challenges that all Digital Earth systems face is the sheer immensity of the amount of data available. These data sets, which are not small in general, are captured over time and regularly kept for posterity. Petabytes worth of geospatial data already exist, and this amount grows every day at an increasing rate due to improving fidelity in data capture technologies. Appropriate methods (e.g. multiresolution representations) are needed to handle such large volumes of data and are well-studied topics for each data type.

In the following section, we present a glimpse of the important related works that study these data sets and their applications more comprehensively. We give partial focus to fundamentals such as multiresolution representations and visualization rather than data processing and analysis, which are generally application-dependent.

2.1. Imagery Data Sets

Geospatial imagery data sets, typically categorized into aerial or satellite photographs, are very useful for the visualization and analysis of locations. They are often used as textures for the cells of the discretized Earth at multiple resolutions, but can also be used for analysis or to provide interesting views of the Earth (e.g. spherical panoramic views).

Aerial photographs are photographs taken from aircraft such as helicopters, balloons, etc that do not have a fixed support on the Earth, while satellite photographs are taken by satellites. Aerial images generally benefit from higher resolution and image quality in comparison to satellite images, but are subject to air traffic and other restrictions, while satellites are operational throughout the year and can frequently re-visit locations for time-lapse captures [8].

On the Digital Earth, multiple images of a single region can exist from different times, at different scales, under different projections, with different spatial extents and orientations, and with different viewpoints and lighting; and providing a meaningful connection between these images has become an important subject of research [9] (see Figure 2 (a) and (b)). One powerful tool for image representation at multiple scales is the mip-map — a pyramidal structure consisting of progressively lower resolution versions of a given image [10] — which has been used in [11] to establish a continuous transition between correlated images at different resolutions. The work of [12] proposes a smooth transition for large scale changes in multiresolution images by blurring the boundary of the high and low resolution images, while in [13], a GPU-centric approach is presented that allows raster data with overlaps and differences in resolution to be combined on the fly.

Assignment of data to the cells of a DGGS is most commonly performed via rasterization [14], where each cell of the DGGS is treated as a pixel-like entity and can be assigned attributes such as color or height. In the general case, where the area represented by a pixel in an image does not exactly match that of an area represented by a DGGS cell, the pixel's color can be assigned to the DGGS cell whose area provides the closest match.

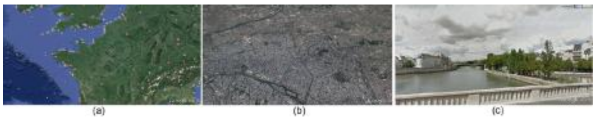


Figure 2: (a), (b) A region of the Earth is captured in two images at different scales. (c) A spherical panoramic view from Google Earth. Images taken from Google Earth [5].

2.2. *Elevation Data Sets*

Data sets that describe terrain on the surface of the Earth generally come in one of two forms. Digital Elevation Models (DEMs), also known as Digital Terrain Models (DTMs) or height maps, are regular grids of terrain elevation values (Figure 3 (a)), whereas Triangulated Irregular Networks (TINs) are sets of triangular faces with arbitrary connectivity (i.e. polygonal meshes) with elevation values stored in the triangle vertices as coordinates (Figure 3 (b)).

The most common form of elevation data, DEMs, are raster data sets and can be processed using image processing techniques and assigned to the cells of a DGGS in much the same way as imagery data (i.e. by assigning a height attribute to each cell). Conversions from TINs to DEMs are possible and can be used to represent TIN data on a DGGS.

The acquisition of elevation data can be accomplished using several techniques. Ground surveying, though expensive and time consuming, is very accurate. It determines the 3D locations of points on the Earth via measuring the angles and distances from that point to known reference locations. Laser scanning using LIDAR technology on a low altitude aircraft [15] can be used to obtain points clouds of the 3D terrain. Photogrammetry — the process of deriving measurements from imagery data [16] — provides another option.

In addition to real elevation data sets, terrain models can also be synthesized through interactive techniques [17, 18, 19] or by adding details using fractals, Perlin noise, or wavelets [20, 21, 22]. These synthetic elevation data sets are usually used for games, simulations, or aesthetic design and have limited use in real geospatial data analysis.

The modeling, rendering, compression, and multiresolution representation of such models are very well studied, and survey papers exist that discuss these subjects for terrains in detail [23, 24, 25, 26, 27]. As discussed in [28, ch. 3], visualization approaches include creating a triangle mesh, displacement mapping, and ray tracing.

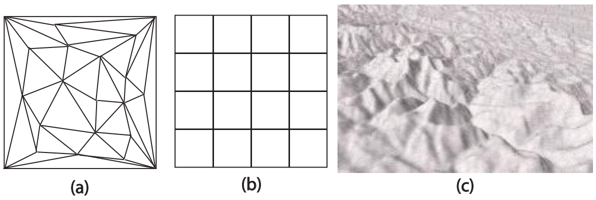


Figure 3: (a) Arbitrary connectivity in a TIN. (b) Regular (grid) connectivity in a DEM. (c) Elevation data sets represented on a Digital Earth. Image taken from [29].

2.3. *Vector Data Sets*

Vector data sets are very important within a Digital Earth system. These data sets — points, polylines, or polygons on the Earth — are comprised of points connected by spherical or ellipsoidal arcs and can represent region boundaries (e.g. nation or city boundaries) or geospatial networks and features (e.g. road and river

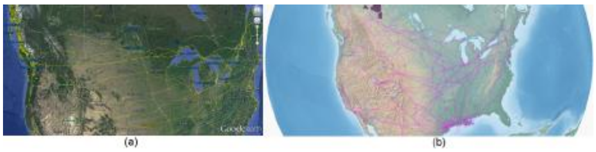


Figure 4: Vector data sets as visualized in (a) Google Earth [5] and (b) PYXIS WorldView [6].

networks or residential house outlines). Figure 4 illustrates some vector features.

In order to store vector data in a DGGS, either rasterization or a bucket-based approach can prove useful [14]. Under the rasterization approach, the vector can be represented as an ordered set of cells, where each cell corresponds to and contains one of the vector's vertices/points. Treating the cells of the DGGS as buckets allows data storage techniques similar to those used by quadtrees to be employed.

As with elevation data sets, generation of vector data can be accomplished with ground surveying, LIDAR, and photogrammetry [30]. Features in LIDAR data sets can be detected using automatic or semiautomatic techniques and vectorized [31, 32, 33, 34], and can be combined with imagery data to enhance the quality of the obtained features [35]. Imagery data alone has been used extensively with photogrammetric techniques to extract building and road contours [36, 37, 38, 39], for which computer graphics techniques such as snakes and edge detection can be quite useful [40, 41].

Additionally, techniques and methods exist to synthesize and edit vector data sets. Sketch-based systems [42, 43, 44, 45] and procedural modeling [46, 47, 48, 49] are common approaches. The work of [50], in particular, discusses applications of procedural modeling to generating a variety of data sets for a virtual world.

As a core form of geospatial data, much study has gone into the visualization of vector data on terrains [51, 52, 53] and they may be found on many Digital Earths [54, 55, 5, 6]. Example visualizations from Google Earth and PYXIS WorldView are shown in Figure 4.

The work of [55] categorizes the various approaches to visualization into three main groups (see Figure 5). Texture-based approaches rasterize the vector data into textures that are then mapped onto the terrain surface [56]. Geometry-based approaches (often used for the simultaneous representation of digital elevation data and vector data [52, 57, 58, 59, 60]) allow the geometry of the vector data to exist separately from the geometry of the terrain, albeit modified for consistency with the terrain shape [61]. Shadow volume-based approaches, such as that described in [51], extrude the vector geometry into polyhedrons that are then rendered into the stencil buffer to distinguish between visible and invisible parts of the scene.

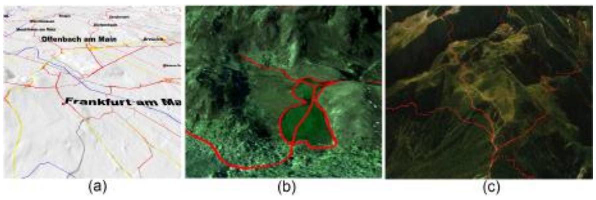


Figure 5: (a) Texture-based; Image taken from [56], (b) Geometry-based; Image taken from [52] and (c) Shadow volume-based vector representations; Image taken from [51].

2.4. *3D Geometric Data Sets*

The current trend of Digital Earth systems shows that their visualization aspects (and perhaps analysis and processing in the near future) can be improved via the inclusion of another type of data: 3D geometric data sets [62, 63] that model objects such as houses, office towers, bridges, cars, or trees. Such 3D models may be created using interactive modeling techniques or by the use of automatic methods, which can generate models based on large sets of images or LIDAR point cloud data [64, 65, 62]. Various methods and techniques for the construction of 3D geometric models from available point cloud or image data may be found in [66].

Aside from Google Earth, which incorporates 3D models into its framework (see Figure 6), few Digital Earths make use of 3D geometric data sets.

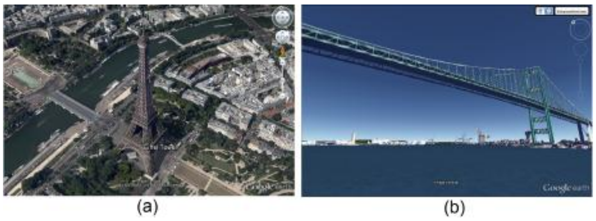


Figure 6: (a) The Eiffel tower in Paris. (b) The Vincent Thomas Bridge in Los Angeles. Images taken from Google Earth [5].

2.5. *Statistical and Other Data Sets*

The data sets mentioned above — elevation, imagery, vectors, and 3D models — represent physical locations on the Earth and are very important for a Digital Earth system. Additional, and equally important, data come in the form of quantitative data sets (represented in numerical or string format) that describe aspects of locations besides their physical properties. Environmental data, biological data, population count, average family or individual income, and a particular location's name are just some of the data that may be assigned as attributes of a spatial data set.

Often these data are statistical in nature and are gathered by randomly sampling regions of interest through various means, such as sensors, surveys, observations, and reporting [67, 68]. Therefore, sampling the Earth efficiently via an appropriate choice for cell shape (see Section 3.1) is important, and different methods for designing such samplings have been suggested [69, 70]. In the next section, we discuss the processes by which DGGS cells are created, for which sampling efficiency is just one of many considerations.

3. Discrete Global Grid Systems

A Digital Earth system provides a multiresolution representation of the Earth as a spatial reference model with the ability to embed, visualize, retrieve, and analyze data at different levels of detail [71]. Underlying the majority of these systems is a structure known as a Discrete Global Grid System (DGGS) — proposed in [71, 72] — that partitions the Earth into a hierarchy of mostly regular cells.

The traditional approach to discretizing the Earth is to use the latitude/longitude coordinate system on a sphere [72]. Here, the 2D domain (or planar map of the Earth) is partitioned into a grid of cells by taking incremental steps along the latitudes and longitudes. These planar cells may be further divided/refined for increased resolution and mapped to the sphere via the well-known spherical coordinate equations noted in Section 3.4. The resulting spherical cells are mostly regular and quadrilateral in shape, with singularities and triangular cells appearing at the poles and the size of the cells varying over different latitudes (see Figure 7).

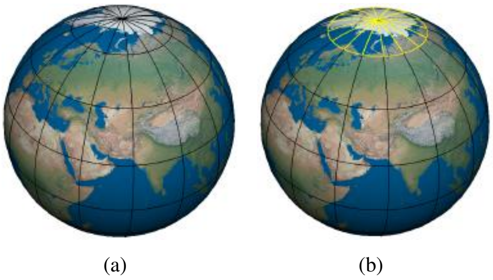


Figure 7: (a) The Earth discretized into quadrilateral cells using lat/long coordinates. (b) Cells close to the pole are triangular.

This approach can be generalized and used to obtain (mostly regular, or *semiregular*) cells with more uniform shape and size. Beginning with an initial discretization of the Earth into planar cells (often 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 Figure 8). Given a regular refinement, a multiresolution hierarchy between cells (in which each cell has a coarser resolution parent and a number of finer resolution children) with a semiregular cell structure can be systematically defined.

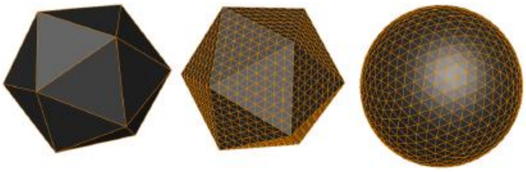


Figure 8: Illustration of the generalized approach to Earth discretization. An initial discretization (here provided by an approximating icosahedron) is refined and projected to the Earth in order to form spherical cells.

Once the cells of a DGGS are created, data can be assigned to them. For this purpose, a mechanism for referring to individual cells is required. Indexing methods on the cells support data assignment and retrieval and the handling of essential queries.

This generalized approach allows different DGGSs to be distinguished from each other in terms of the following elements: the initial discretization and cell shape, the cell refinement method, the projection method, and the employed indexing method. Here, we describe many of the different possibilities for each of these elements of a DGGS, plus topics concerning the rendering of a DGGS towards the end of the section.

3.1. Cell Shape

The core of any DGGS is its cell structure. Each cell can be considered to be a (spherical) polygon on the surface of the Earth, for which several different cell shapes — triangles, quads, or hexagons — can be used. Each cell shape has its own advantages and disadvantages that impact the ease with which operations may be applied to the geospatial data.

As they work well together with Cartesian coordinates and lat/long discretizations, quadrilateral cells are fairly easy to use. DGGSs and spherical representations that have employed them include the works of [73, 74, 75, 76, 28, 77].

Triangular cells are also simple to use, in addition to being efficient to render and compatible with the faces of polyhedrons such as the icosahedron, dodecahedron, or tetrahedron that are often used to approximate the Earth (see Section 3.2). For example, triangular cells are used by Dutton [78] with the octahedron and Lee and Samet [79] with the icosahedron.

As discussed by Sahr [80], hexagonal cells are preferable in some applications due to uniform adjacency [81], regularity, and efficiency in sampling [82, 83], despite greater challenges in refinement and indexing. As a result, hexagons are widely employed in representations of the Earth [14, 6, 84, 85, 86, 87].

The cell shape can arise naturally from the chosen initial discretization and refinement method, or can be targeted via specific choices of initial discretization and refinement.

3.2. *Initial Discretization*

Discretizing the Earth into a initial set of planar cells is an important first step for any DGGS. Aside from methods that partition a 2D map of the Earth, of which latitude/longitude grids are the standard, the Earth may be discretized quite easily using an initial approximating polyhedron. DGGSs that make use of initial polyhedrons are known as Geodesic DGGSs [72]. Among the most common choices for an initial polyhedron are the platonic solids — the tetrahedron, cube, octahedron, dodecahedron, and icosahedron — and the truncated icosahedron.

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 [88, 78, 89, 84, 90], whereas the simplicity of the tetrahedron has motivated its use in [28]. The cube benefits from alignment with the Cartesian coordinate system and the prevalence of compatible data structures, such as quadtrees [91, 92, 93, 94].

While the dodecahedron features lower angular distortion under equal area projection (such as Snyder projection [95]; see Section 3.4) compared to the other platonic solids [96], 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 [97, 14, 88, 98, 85]. 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 [95, 70].

Generally, the faces of these polyhedrons cover large areas of the Earth's surface. In order to increase the resolution of the DGGS, refinements may be employed.

3.3. *Refinement*

Refinement methods are methods that produce a set of fine cells from a set of coarse cells in a uniform and predictable manner, and can be used to construct more cells in a Geodesic DGGS by introducing more faces into an approximating polyhedron.

Refinement methods are categorized according to their input cell shape(s), output cell shape (typically the same as the input shape), and *aperture* or *factor* (the ratio of the area occupied by a coarse cell to that occupied by a fine cell). For instance, the hexagonal 1-to-3 refinement shown in Figure 9 (b) takes coarse hexagonal cells and produces fine hexagonal cells that have $\frac{1}{3}$ the area of the coarse cells. The fine cells produced by a refinement may be assigned to coarse cells as *children*, defining a hierarchy between cells. In addition to the 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, and a DGGS is called congruent when its employed refinement is congruent. For example, quadrilateral 1-to-4 refinement (as shown in Figure 9 (a)) is congruent while hexagonal 1-to-3 refinement is *incongruent* (see Figure 9 (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 9 (c)).

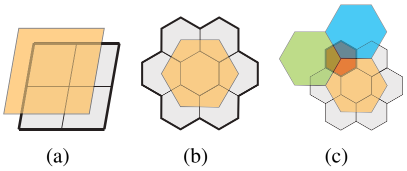


Figure 9: (a) Quadrilateral 1-to-4 refinement is a congruent refinement. (b) 1-to-3 hexagonal refinement is not congruent. (c) A child can have several potential parents under an incongruent refinement. The red child hexagon, for instance, has three potential parents.

Alignment refers to the case in which a coarse cell shares a particular point in common with a fine cell. If the cells share a vertex, the refinement and any DGGS that employs it is called vertex-aligned. If the cells share a centroid, they are called center-aligned. Refinements that are neither vertex-aligned or center-aligned exist, but are fairly uncommon. As we are primarily concerned with center-alignment over vertex-alignment, we will refer throughout the paper to refinements and DGGSs that are center-aligned as *aligned*, and those that are not as *unaligned*.

Center-alignment in a DGGS is quite a beneficial property to have, as it implies that traversing from one resolution to another always results in an improvement in accuracy. Accuracy, here, is a measure of how well a point p 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 10 illustrates a comparison between unaligned quadrilateral 1-to-4 refinement and aligned 1-to-2 refinement.

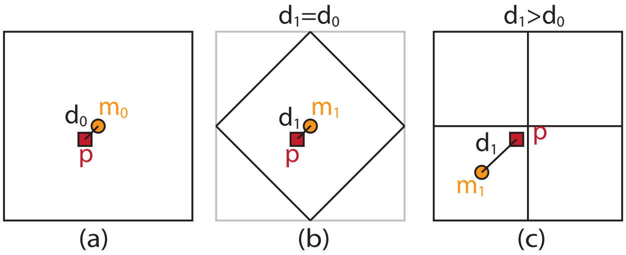


Figure 10: Accuracy is measured in terms of the distance d_r from a point p to the centroid m_r of its containing cell through the resolutions. (a) d_0 is the distance between m_0 and p, illustrated as a red square. (b) Under center-aligned refinements, the accuracy is at least as good as that of the previous resolution. (c) Under refinements that are not center-aligned, the accuracy may be decreased (d_1 ¿ d_0).

In addition to these properties, a low 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.

In general, it is difficult for a refinement to exhibit all of these desirable properties. For example, it is possible to define an aligned 1-to-2 refinement for quadrilateral cells which has the slowest rate of growth (Figure 11 (a)) [94], but which is not congruent, while the commonly employed 1-to-4 refinement for quadrilateral cells (Figure 11 (b)) is congruent but unaligned [91, 93]. Quadrilateral 1-to-9 refinement is congruent and aligned but has a larger factor (Figure 11 (c)) [92]. Triangular cells in DGGSs tend to be refined by a congruent and aligned 1-to-4 refinement (see Figure 11 (d)) [78, 89, 99].

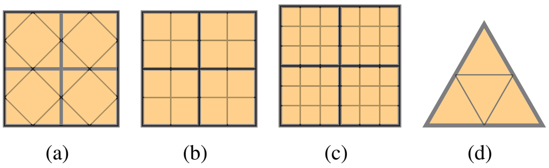


Figure 11: (a) An aligned 1-to-2 refinement that is not congruent. (b) A congruent 1-to-4 refinement that is unaligned. (c) An aligned and congruent 1-to-9 refinement. (d) An aligned and congruent 1-to-4 refinement for triangles. The boundaries of coarser cells are highlighted using thicker lines.

Aside from quadrilateral and triangular cell refinements, there exist a number of types of hexagonal refinement [29], each of which exhibits characteristics that can prove useful in DGGS applications and in which growing interest has developed [87, 100]. For example, Sahr [14] uses hexagonal 1-to-3 refinement, which has the lowest factor possible for hexagonal refinements, on the icosahedron while Vince [84] uses the same refinement on the octahedron. While this and other hexagonal refinements introduce a rotation in the lattices of two successive resolutions [101], hexagonal 1-to-4 refinement produces rotation free lattices at all levels of resolution, simplifying hierarchical analysis [102, 103]. Two types of 1-to-4 hexagonal refinement can be combined to obtain a better hierarchy, as performed by Tong et al. [104], 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.

Refinement methods are also used in the subdivision methods studied in computer graphics; but while subdivision methods alter the geometry of the polyhedral faces via smoothing masks, refinements on DGGS cells have no such impact.

3.4. Projection

For a DGGS, cell geometry results from an employed projection method, where planar cells and their assigned data may be projected onto the surface of the Earth using one of a variety of spherical projections. Such projections have a rich history in the field of cartography, and have been used to represent the spherical Earth on flat maps [105, 106, 107]. In a DGGS, they are used to establish a correspondence between the planar cells resulting from refinement on an initial discretization and spherical cells that partition the 3D Earth.

Cartographic Projection: Traditional cartographic projections are transformations from a point on the Earth to a point on a 2D map and can be represented as a function: $p_m = F(p_s)$, where p_s lies on the sphere (or ellipsoid, in the case of ellipsoidal projections), and p_m is located on the 2D map (see Figure 12). The inverse projection F^{-1} defines a mapping from a point on the 2D map to a point on the sphere/ellipsoid and allows flat shapes, such as polyhedral faces, to be mapped to spherical DGGS cells.

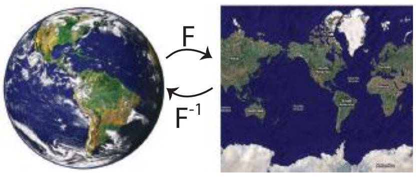


Figure 12: F maps points on the Earth (a sphere) to a map (a 2D domain) while F^{-1} maps points on the 2D map to the sphere. Images taken from [108].

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 $0 \leq \phi \leq \pi$) and longitude (given by fixing angle $0 \leq \theta \leq 2\pi$) serving as the two main axes of the 2D domain. This 2D domain and the sphere are related through

$$F(x,y,z) = \begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} tan^{-1}(\frac{y}{x}) \\ cos^{-1}(\frac{z}{R}) \end{pmatrix} \text{ and } \quad (1)$$

$$F^{-1}(\theta,\phi) = \begin{pmatrix} R\cos(\theta)\sin(\phi) \\ R\sin(\theta)\sin(\phi) \\ R\cos(\phi) \end{pmatrix}, \quad (2)$$

where $R = \sqrt{x^2 + y^2 + z^2}$ is the radius of the sphere (see Figure 13).

Note that, as demonstrated in Figure 13, regular quadrilaterals in the latitude/longitude domain are mapped to quadrilateral cells on the sphere with variable areas (i.e. areal distortion) and singularities at the poles. In general, for a given function

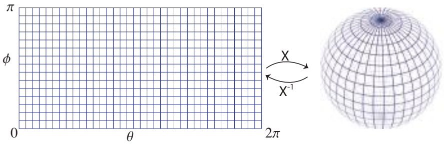


Figure 13: A 2D domain and its associated sphere.

$$G(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}$$

that maps $(u,v) \in \Omega \subset R^2$ to a continuous surface $S \subset R^3$, we can study the distortion behavior of G using its first fundamental form, which provides a sense of the local curvature of a surface [109]. The first fundamental form of G is a matrix of functions $I(u,v) = J(u,v)^T J(u,v)$, where $J(u,v)$ is the Jacobian of G. Using the singular values $\sigma_1(u,v)$ and $\sigma_2(u,v)$ of I, the distortion behavior of G can be evaluated (v, here, is a constant) [110, 83]:

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

For instance, the inverse spherical coordinate function F^{-1} defined in Equation 2 has the first fundamental form

$$I(\theta,\phi) = \begin{pmatrix} R^2\sin^2(\phi) & 0 \\ 0 & R^2 \end{pmatrix},$$

whose singular values are $\sigma_1(\theta,\phi) = R\sin(\phi)$ and $\sigma_2(\theta,\phi) = R$. These singular values satisfy none of the above properties, hence spherical coordinate conversion is neither stretch-, angle-, distance-, nor area-preserving.

Since the sphere is not developable, it is impossible to define an isometric mapping between the sphere (or ellipsoid) and a 2D map [111]. As a result, several types of spherical projection have seen use in the Digital Earth setting, each of which exhibits different properties/distortions. The most common of these are azimuthal, conic, and cylindrical projections (see Figure 14).

Azimuthal projections, such as Gnomonic projection and Lambert's azimuthal equal area projection, map points on the sphere to a plane tangent to the sphere at some point p (Figure 14 (a)). In this type of projection, great circle arcs passing through p are mapped to line segments on the tangent plane. Equidistant azimuthal projections (i.e. those in which the lengths of the great

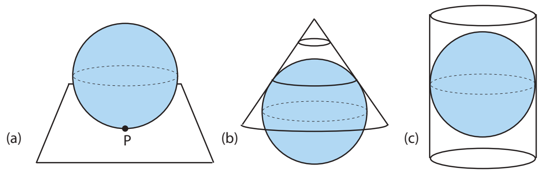


Figure 14: (a) Azimuthal projections map points between the sphere and a given tangent plane. (b) Conic projections map points between the sphere and an intermediate cone. (c) Cylindrical projections map points between the sphere and an intermediate cylinder.

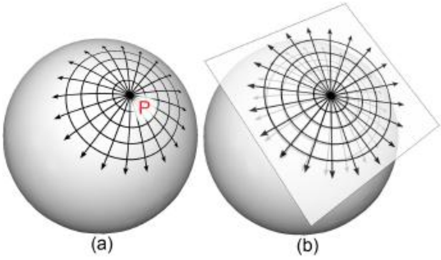


Figure 15: In exponential maps, great circle arcs passing through a point p on the sphere have the same length and direction as line segments passing through p on the tangent plane. Image reproduced from [113].

circle arcs and the line segments are equal) are equivalent to exponential maps (see Figure 15) [112], which are quite useful in computer graphics and have applications in texture mapping, spherical multiresolution, and spherical splines [113, 114, 115].

As the lowest amounts of distortion in azimuthal projections are found closest to the center of the tangent plane, the location of the tangent plane can be adjusted to concentrate unwanted distortions towards less significant locations, as was done with Gnomonic projection and Lambert's azimuthal equal area projection [107, 106]. Based on the location of the tangent plane used for projection, azimuthal projections may be categorized as polar, equatorial, or oblique (see Figure 16) [109].

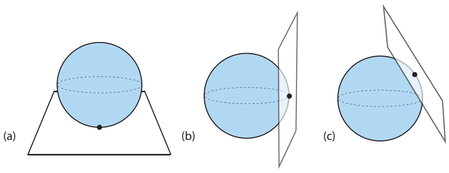


Figure 16: Tangent plane locations in (a) polar, (b) equatorial, and (c) oblique azimuthal projections.

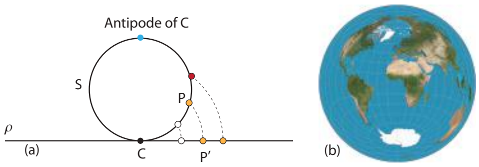


Figure 17: (a) Lambert projection from a sphere S to a plane ρ (b) Image of the 2D Earth after Lambert Azimuthal equal area projection, from Wikipedia.

Among azimuthal projections, Lambert azimuthal equal area projection is particularly important since it forms the base of several equal area projections used in DGGSs. Given a point c on the sphere S and the plane ρ that is tangent to S at c, a point p_s on the sphere is projected to a point p_m on the 2D map by taking p_m to be the intersection of ρ with a circle with center c that passes through p_s and is normal to ρ (see Figure 17). Note that c is projected to itself on ρ and its antipode is excluded from the projection as the intersecting circle is not unique. This projection and its inverse can be explicitly calculated using the following mappings:

$$\begin{aligned} F(x,y,z) &= \left(\sqrt{\tfrac{2}{1-z}}x,\ \sqrt{\tfrac{2}{1-z}}y\right), \text{ and} \\ F^{-1}(a,b) &= \left(\sqrt{1-\tfrac{a^2+b^2}{4}}a,\ \sqrt{1-\tfrac{a^2+b^2}{4}}b,\ \tfrac{a^2+b^2}{2}-1\right). \end{aligned}$$

Conic and cylindrical projections attempt to reduce the high distortion that results from projecting spherical points directly to the plane by first projecting them into an intermediate domain. For conic and cylindrical projections, the intermediate domain takes the form of a tangential cone or a cylinder, respectively (Figure 14 (b) and (c)), both of which may be flattened to a 2D planar map via a straightforward mapping. Examples include Mercator projection (cylindrical and conformal), Lambert's cylindrical equal area projection, and Lambert's conic conformal projection. Just as the location of the tangent plane can be adjusted in azimuthal projections, the orientation of the intermediate cylinder and cone can also be adjusted, resulting in projections such as Transverse Mercator projection [105].

Similar projections that do not fall into these categories — mainly modifications and hybrids of these three types — also exist and have been used in Digital Earth systems. Modified versions include pseudo-azimuthal and pseudo-cylindrical projections. A notable example of a hybrid projection is the one employed in the HEALPix system, which is a combination of Lambert cylindrical equal area projection and Collignon pseudo-cylindrical equal area projection.

Polyhedral Projection: Given a polyhedron as an intermediate domain, it is also possible to define polyhedral globe projections to and from the sphere. Such projections typically operate by dividing the sphere into regions, each corresponding to a face of the polyhedron, and then employing an existing spherical projection to map points between the spherical region and the planar polyhedron face.

Snyder projection [95] — which uses Lambert's azimuthal equal area projection on the faces of the polyhedron and which is defined for all platonic solids plus the truncated icosahedron — is one such polyhedral projection method. Roşca and Plonka's projection [91, 77] uses an intermediate domain for each face to map points between the cube (later generalized to the octahedron in [116]) and the sphere using Lambert azimuthal equal area projection. These projections are illustrated in Figure 18.

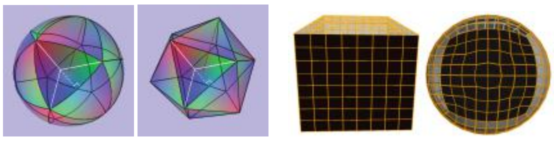


Figure 18: (a) Snyder projection. Image taken from [108]. (b) Roşca and Plonka's projection.

Additional polyhedral globe projections include the slice-and-dice approach [117], in which the projection consists of an equal area partitioning ("slice") step and an equal area positioning ("dice") step; the equal area projection for octahedrons from [118], which slices the sphere into eight sections and establishes a geometric relationship between each section of the sphere and each face of the octahedron; and Fuller's Dymaxion map for icosahedrons [119].

Such projections tend to operate on simple polyhedrons, yet it is also possible to map between the sphere and a polygonal mesh using spherical projections from the field of computer graphics. Though they have not yet been employed on DGGSs, works such as Praun and Hoppe's [120] and Sheffer et al.'s [121] map points between genus 0 polygonal meshes and the sphere for the purposes of mesh parametrization and remeshing. Praun and Hoppe's approach attempts to minimize stretch distortion while Sheffer et al.'s approach attempts to minimize angle or area distortion. For more information on mesh parametrization, see [110].

Equal Area Projections: Though spherical projections can be stretch-preserving, conformal, or equal area, those that are equal area tend to be best fit with DGGSs, as noted by White et al. [70], since these systems use planar or piece-wise planar (polyhedral) domains to sample the spherical surface of the Earth. As a result, they are widely used in DGGSs. Examples of equal area cartographic projections include Lambert's cylindrical, Lambert's azimuthal, Mollweide's, and Werner's equal area projections (see Figure 19) [106, 108].

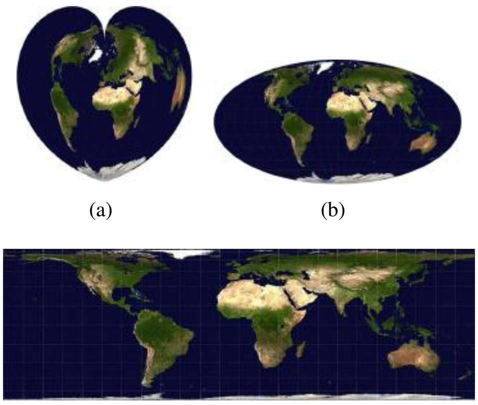


Figure 19: (a) Werner, (b) Mollweide, and (c) Lambert (cylindrical) projections. Images taken from [108].

Snyder's and Roşca and Plonka's projections, too, are equal area projections, and Snyder projection in particular has seen extensive use within the Digital Earth community due to its equal area property and low angular distortion. Though Snyder projection has a closed form, its inverse does not and requires an expensive iterative technique to compute. The work of [122, 123] describes methods to speed up this inverse process, whereas the work of [94] uses Roşca and Plonka's method, which has closed forms for both the forward and inverse projection.

Ellipsoidal Projections: In addition to spherical projections, ellipsoidal projections may be defined and characterized similarly (i.e. as azimuthal, conic, or cylindrical and as stretch-preserving, conformal, or equal area). Such projections are common in Earth representations, as an oblate spheroid provides a better approximation of the shape of the Earth than the sphere [124, 125].

An oblate spheroid is an ellipsoid with two semi-principal axes of equal length ($a = b$) and one semi-principal axis of smaller length ($c < a, b$). Taking $a = b = 6,378,137$ m and $c = 6,356,752.3142$ m [28] results in an ellipsoidal representation of the Earth known as the World Geodetic System 1984 (WGS 84) ellipsoid. Though other ellipsoidal representations for the Earth exist, e.g. Bessel's ellipsoid, the WGS 84 ellipsoid is currently standard and is used in many virtual globe and GIS applications, such as in Google Earth

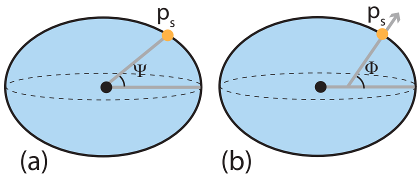


Figure 20: (a) Geocentric latitude ψ is calculated with respect to a vector from the center of the ellipsoid to p_s. (b) Geodetic latitude ϕ is calculated with respect to the surface normal at p_s.

[126] and GPS [127].

In general, ellipsoidal projections are more difficult to define than spherical projections. As with the sphere, a simple solution is to use a latitude/longitude discretization of the ellipsoid. In the case of an oblate ellipsoid such as the WGS 84 ellipsoid, the calculations for longitude θ are the same as for the sphere. Latitude, however, may be defined as either geocentric or geodetic. For a point p_s on the ellipsoid, its geocentric latitude ψ (see Figure 20 (a)) is the angle that a vector from the center of the ellipsoid to p_s makes with the equatorial plane, whereas geodetic latitude ϕ (see Figure 20 (b)) is the angle that the surface normal at p_s makes with the equatorial plane [105].

Another simple way to define an ellipsoidal projection is to first define a projection between the ellipsoid and the sphere and then employ a pre-existing spherical projection. Web Mercator projection (see [128]) is an example of such a projection and is used by most major online map applications, including Google Earth [129]. To project a point p_s on the ellipsoid to a point p_m on a map, the Web Mercator projection first converts p_s to a point s on the sphere using its longitude θ and geodetic latitude ϕ (say, $s = F^{-1}(\theta, \phi)$ where F^{-1} is from Equation 2), and then applies spherical Mercator projection to s (see Figure 21). Note that this conversion from p_s to s is nonconformal, hence Web Mercator is neither conformal nor equal-area. Conformal and equal-area projections between the sphere and the ellipsoid can be found in [105].

Generalizations of spherical projections to ellipsoids are also possible. Gauss and Krüger generalized Transverse Mercator projection to the ellipsoid (as a result, it is sometimes known as Gauss-Krüger projection) [105], which forms the basis of one of the most important ellipsoidal projections: Universal Transverse Mercator (UTM) projection. UTM projection divides the Earth into sixty vertical strips along lines of longitude, and employs (ellipsoidal) Transverse Mercator projection on each of these strips such that distortion is minimized lengthwise.

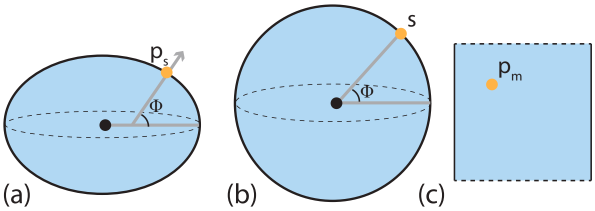


Figure 21: Web Mercator projection. (a) A point p_s on the ellipsoid has geodetic latitude ϕ. (b) A point s on the sphere is found by using ϕ as the spherical latitude. (c) The point s is projected to planar point p_m via Mercator projection.

In a way, UTM projection bears some similarities with polyhedral projections, which may also be defined for ellipsoids. The work of [124], in which the ellipsoid is projected to a cube, provides one such projection. Note, however, that ellipsoidal polyhedral projections have not been as extensively studied as spherical polyhedral projections, due to the difficulty of defining them.

Whichever projection is chosen, the combination of an initial discretization, a refinement method, and a projection method allow a multiresolution hierarchy of DGGS cells to be generated in a systematic manner, for which a systematic mechanism for referring to individual cells is required.

3.5. Cell Indexing

A core functionality of any DGGS is assigning and retrieving data to and from the cells that discretize the Earth. While this can be accomplished using data structures such as quadtrees (the method of choice for establishing efficient access to hierarchical data in other problem domains) as in [91, 97], the sheer immensity of the Digital Earth causes tree structures that record node dependencies to become prohibitively expensive.

Hence, the role of providing data access in a DGGS is often served by cell indexing methods. For each cell that exists in a DGGS, an indexing method assigns to it an index i that uniquely identifies that cell. A cell's index i can be used to determine that cell's location on the Earth and can serve 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, most indexing methods for DGGSs are derived using three general indexing mechanisms: hierarchy-based, space-filling curve (SFC) based, and coordinate-based. A more comprehensive overview of DGGS cell indexing methods and conversions between them may be found in [130].

Hierarchy-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 or postfix for its own index. Formally and without loss of generality, if cell c at resolution r has index $Id_0d_1d_2\ldots d_{r-1}$, its children f_i receive indices $Id_0d_1d_2\ldots d_{r-1}i$.

Note that if the maximum number of children per cell is b, then the digits d_j lie in the range $[0, b-1]$ 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 [85, 72, 89].

Figure 22 (a) illustrates the hierarchy-based indexing proposed in [131] for the quadrilateral cells obtained from 1-to-4 refinement. The SCENZ-Grid system [92] employs a similar indexing method for the quadrilateral faces of a cube under aligned 1-to-9 refinement (Figure 22 (b)).

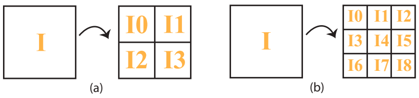


Figure 22: (a) Hierarchy-based indexing for 1-to-4 refinement. (b) Hierarchy-based indexing method for aligned quadrilateral 1-to-9 refinement.

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 (see Figure 23 for an example). Works on DGGSs that have made use of hierarchy-based indexing include [93] (quadrilateral cells), [78, 99, 89] (triangular cells), and [86, 85, 14] (hexagonal cells).

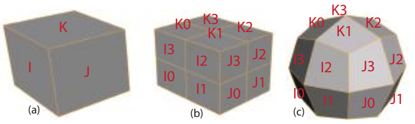


Figure 23: Hierarchy-based indexing given an initial polyhedron. (a) The faces of the initial polyhedron are indexed using letters. (b) The faces that result from refinement inherit the parent index. (c) Faces are projected to the sphere.

Benefits of hierarchy-based cell indexing methods include efficient hierarchical traversal of the cells and the ability to determine a cell's resolution directly from the index's length.

Curve-based: Another means of indexing cells, known as curve-based indexing, is to run a curve through all the cells of a given resolution (see Figure 24) and index them according to the order in which they are intersected. These curves may be denoted by $f(t) : T \subset R \rightarrow Q \subset R^2, t \in T$, with examples including row-major traversal and Morton (Z) curves.

As all the cells in the space must be visited, Space Filling Curves (SFCs) — such as the Hilbert, Peano, and Sierpinski curves — are sensible choices for defining such indexings. SFCs are recursively-created curves whose range covers an entire space, and are surjective and continuous mappings from $T \subset R$ to $Q \subset R^2$. The base curve is typically defined via a simple initial geometry on a simple domain, which is then recursively refined and the simple geometry repetitively transformed to cover the entire refined domain as governed by a set of production rules.

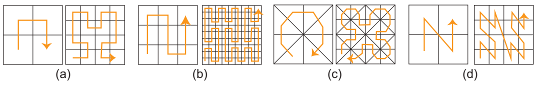


Figure 24: (a) Hilbert, (b) Peano, (c) Sierpinski, and (d) Morton curves.

Typically, if the initial geometry of the SFC covers i cells of the initial domain, a 1-to-i refinement is suitable for application on the domain and may be associated with the SFC. This association makes it natural to define a base b for the indexing method (usually taken to be $b = i$ or $b = \sqrt{i}$ if 1-to-i refinement is associated with the curve), which allows the resolution of the cell to be determined directly from the length of its index. 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 $2r$.

For instance, the refinement associated with the Hilbert and Morton curves is quadrilateral 1-to-4 refinement, as both are defined on initial two by two quadrilateral domains. Therefore, indices of base four or two are appropriate for Hilbert and Morton curves (see Figure 25).

One of the main benefits of curve-based indexing is to out-of-core algorithms that operate on data that cannot fit into main memory, such as in the work of [132]. Contiguous cell regions obtain similar indices and, as a result, can be stored in contiguous sections of external memory. This results in fewer I/O operations when performing range queries.

Works that have made use of curve-based indexing for DGGSs include [133, 88], which used Morton indexing on cells resulting from 1-to-4 refinements on the icosahedron and octahedron; and [134], which used the Sierpinski SFC to index triangular cells refined with a factor of two. See Figure 26 for an illustration of [88]'s method on an unfolded octahedron.

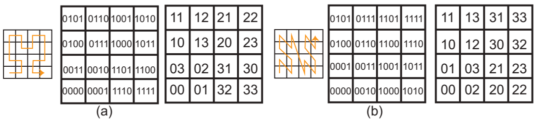


Figure 25: (a) Hilbert-based indexing in base 2 and base 4. (b) Morton-based indexing in base 2 and base 4.

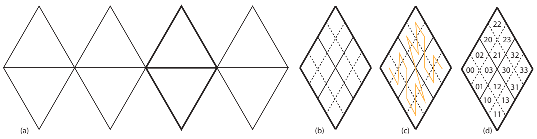


Figure 26: Morton indexing on the octahedron. (a) Unfolded octahedron. (b) Two triangles are merged to obtain a diamond, which is then recursively refined. (c), (d) Morton SFC-based indexing on the diamond.

Coordinate-based: 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 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 $(i_1, i_2, ..., i_m)_r$, 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 27 (a). Another, seen in [84, 90], is a 3D coordinate indexing system in which the Barycentric coordinate of each cell is taken to be its index. In [135], 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 [136, 29]. In this case, indices are augmented with additional information in order to specify the initial face to which a coordinate value belongs. Hence, a cell $(i_1, i_2, ..., i_m)_r$ in the coordinate system of face f is given index $[f, (i_1, i_2, ..., i_m)_r]$.

Taken together, these elements — cell shape, initial discretization, refinement, projection, and indexing — allow one to define a Discrete Global Grid System, which may then be visualized on a screen.

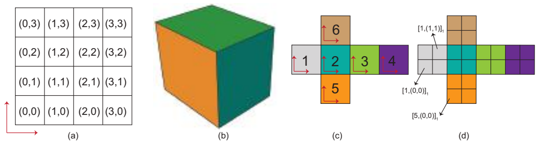


Figure 27: (a) Coordinate-based integer indexing based on Cartesian coordinates. (b) A cube. (c) The cube unfolded with coordinate systems assigned to each face. (d) The indices of some cells are shown after one step of 1-to-4 refinement.

3.6. DGGSs in Rendering

The visualization of digital globes presents a number of unique and interesting challenges that, mostly owing to scale, differ significantly from other related visualizations, such as virtual worlds in video games. While Digital Earths and video games both deal with large amounts of data and have requirements on accuracy and reliability, Digital Earths must contend with a far greater amount of data (and at greater scale) and more demanding requirements.

A comprehensive overview of virtual globe rendering can be found in [28], [137], and, to a lesser extent, [138] along with a summary of several of these challenges and techniques to deal with them. In many works on virtual globe rendering (e.g. [28, 139, 140, 124, 138, 137]), multiresolution tessellations of the Earth into mostly regular cells (i.e. DGGSs) serve as the backbone for addressing these challenges.

For example, as a Digital Earth should accommodate views at the scale of the entire globe and views at the scale of street level in a city, 32- and even 64-bit floating-point precision are often inadequate and can result in depth testing and jittering artifacts. Solutions for depth buffering artifacts include adjusting the near and far plane positions, using complementary depth buffering, using logarithmic depth buffering, and using multiple view frustums [28, 137]. In the case of jittering, one can render objects using relative — rather than absolute — coordinates, as described in [141, 142].

As the cells of a DGGS cell hierarchy exist at different scales (see Figure 28), a sensible approach is to specify these coordinates relative to an appropriate cell. The Ellipsoidal Cube Map (ECM) of [124], for instance, is able to achieve decimeter and centimeter resolutions across the Earth by using camera-centric relative coordinates for the corners of the DGGS quad cells, and quad-centric relative coordinates for the data assigned to each quad cell, when rendering.

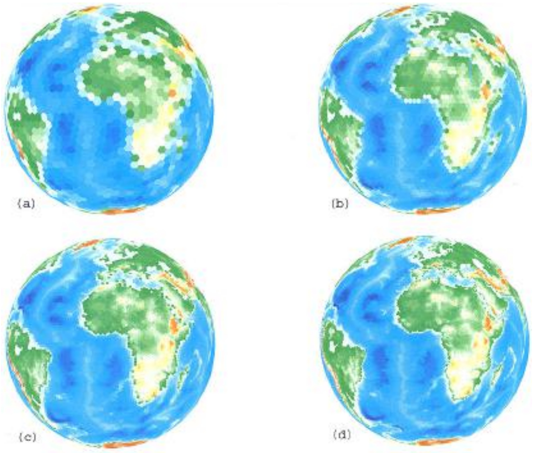
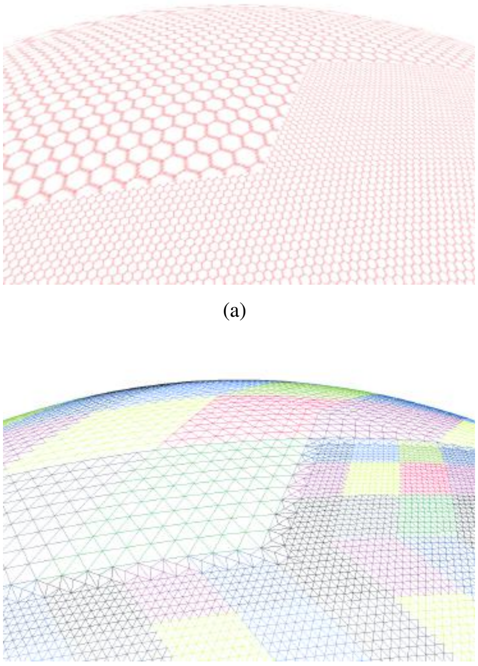


Figure 28: The multiresolution cell hierarchy of a DGGS allows the Earth to be analyzed and visualized at different scales. Image taken from [72].

Another and very important challenge is that of dealing with the massive amount of data stored in a Digital Earth. As geospatial data sizes can measure in terabytes, some amount of pre-processing and simplification (such as clip-mapping [143, 144, 145], a modified version of mip-mapping; see the entry on Google Earth in Section 4) is required to fit such data into memory and render it in a reasonable amount of time. One of the main advantages of a DGGS in this sense is that a DGGS's multiresolution cell hierarchy supports view-dependent level-of-detail rendering (see Figure 29), which has some relation to the cartographic process of map generalization. An extensive discussion on how Cesium's DGGS, in combination with view volume and horizon culling, was used to handle massive data amounts can be found in [137].

An important aspect of level-of-detail rendering is in dealing with gaps that may appear between cells rendered at high resolution and cells rendered at a lower resolution, particularly in the case of incongruent refinements. This is the same problem faced by adaptive subdivision techniques within the field of computer graphics, for which various solutions have been proposed; including creating a set of transitionary faces between the high and low resolution faces, as in [146], or splitting low resolution faces in order to accommodate the higher number of vertices of the high resolution faces, as in [147]. A blur function can also be used to provide a transition between the high and low resolution cells [138].

Finally, a third challenge lies in rendering a curved Earth when graphics hardware is designed for planar shapes such as triangles and quads (see Figure 30 (a)). A tessellation of the Earth into a mesh of planar polygons is hence essential, and already provided by a DGGS cell hierarchy. For DGGSs made up of hexagonal cells, conversion to a triangular tessellation via taking the dual of the hexagonal lattice, as discussed in [29], improves the efficiency of rendering.

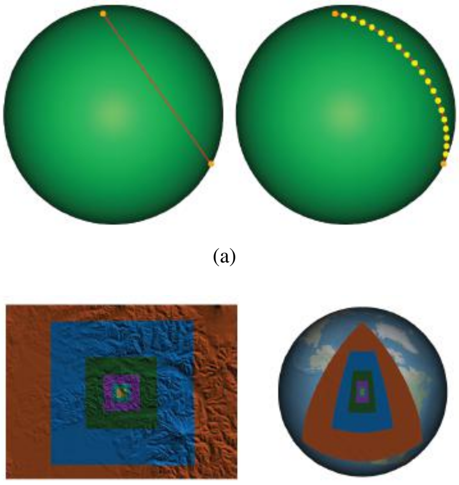

(a)

(b)

Figure 29: View-dependent level-of-detail rendering of a DGGS. Hexagonal cells can be converted to a triangular representation for improved efficiency. Transitional faces bridge the gap between the high resolution and low resolution faces.

However, distortions in the chosen projection can manifest as artifacts in the rendered globe, such as oversampling near the poles and stretching along the equator in the case of latitude/longitude discretization (see Figure 30 (b)). DGGSs with more uniform cell size and distribution can offer improved tessellations. An alternative solution, described in [145] in the context of rendering spherical terrains using clip-maps, is to fix the viewer position atop a static Earth geometry and resample the data about the viewer using spherical coordinates.

Concrete discussions on the implementation of a virtual globe renderer using a DGGS are presented in [140, 138, 137]. In the next section, we provide an overview of state-of-the-art DGGSs currently in use.


(a)

(b)

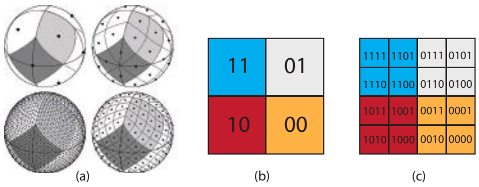
Figure 30: The challenge of curvature. (a) Graphics hardware is designed for planar shapes, hence special care is needed to produce spherical shapes. (b) A clip-map with oversampling at the poles and stretching at the equator due to projection distortions. Images taken from [28].

4. State-of-the-art DGGSs

With the variety of options available for the design of a DGGS, it should come as no surprise that a number of different DGGSs have been proposed in the literature and/or implemented in the industry. We briefly describe a selection of these systems in this section. A list of Digital Earths and visualization applications can be found in [148].

HEALPix: One example of a DGGS is that of HEALPix (Hierarchical Equal Area isoLatitude PIXelation of a sphere), for which a software package has been released for data analysis simulations and visualizations of the Earth [93, 149]. In the HEALPix system, the Earth is approximated using a rhombic dodecahedron (12 initial quadrilateral faces) that are then refined using a congruent 1-to-4 refinement (see Figure 31). 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.

ECM and **SCENZ-Grid**: Another example of a DGGS is the Ellipsoidal Cube Map (ECM) [124], which uses 1-to-4 refinement on the faces of a cube that circumscribes an ellipsoidal Earth. The Quadrilateralized Spherical Cube (QSC) projection was employed in order to minimize areal and angle distortions.


11	01
10	00

1111	1101	0111	0101
1110	1100	0110	0100
1011	1001	0011	0001
1010	1000	0010	0000

Figure 31: The HEALPix system. Image taken from [93]. (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.

The SCENZ-Grid system also starts with a cube, but rather than a congruent 1-to-4 refinement, a congruent and aligned 1-to-9 refinement is applied to establish the multiresolution representation [92]. The projection used is HEALPix's projection, and its indexing method is hierarchy-based and in base 9 (see Figure 32). Mainly used for applications in environmental monitoring, this project is the result of a collaboration between New Zealand's Landcare Research [150] and GNS Science [151].

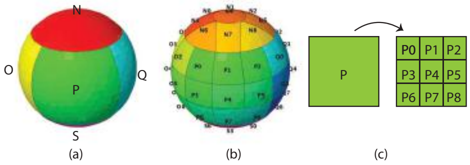

P0	P1	P2
P3	P4	P5
P6	P7	P8

Figure 32: The SCENZ-Grid system. Image taken from [92]. (a) Repeated refinement and projection on the sphere. (b), (c) Quadrilateral cell indexing system.

Crusta: The Crusta DGGS, proposed in [139], was designed to support high-resolution topography data and imagery. Starting from a 30-sided rhombic triacontahedron, 1-to-4 refinement is applied on the quad cells and the resulting vertices are normalized to the geoid.

QTM and **HSDS**: Dutton and Goodchild, two pioneers in designing Digital Earth models, have proposed their own DGGSs.

Dutton's Quaternary Triangular Mesh (QTM) [78] creates cells using congruent and aligned 1-to-4 quaternary refinement on the faces of the octahedron (see Figure 33). 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. The projection used is a specially designed projection called Zenithal Orthotriangular Projection (ZOT), which is neither equal-area nor conformal [152].

Goodchild's Hierarchical Spatial Data Structure

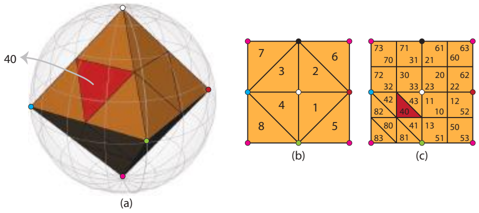


Figure 33: The QTM system. (a) An octahedron embedded in a sphere. (b), (c) The indexing method applied to the unfolded faces of the octahedron.

(HSDS) system [89] also estimates the globe using the octahedron refined under a congruent and aligned 1-to-4 triangular refinement. Unlike many other DGGSs, however, the refinement is applied directly on the sphere (using spherical rather than Euclidean midpoints) after associating each face of the octahedron with a spherical triangle. The implicit projection is neither equal-area nor conformal. The indexing method employed is a slightly modified version of that used by QTM (see Figure 34).

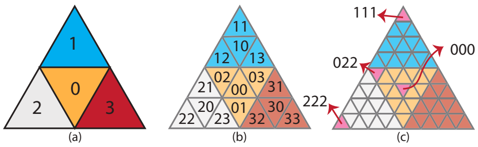


Figure 34: The HSDS system. (a), (b) Hierarchy-based cell indexing. (b) Ordering of the children of each cell.

ISEA3H and **PYXIS Indexing**: A number of DGGSs with hexagonal cells have arisen due to the beneficial properties of hexagonal cells in Earth representation, such as the Icosahedral Snyder Equal Area Aperture 3 Hexagon (ISEA3H) DGGS. As can be inferred from the name, the cells of the ISEA3H DGGS arise from aligned hexagonal 1-to-3 refinement on an icosahedron (see Figure 35), the initial application of which results in a truncated icosahedron, which are projected to the sphere via inverse Snyder projection.

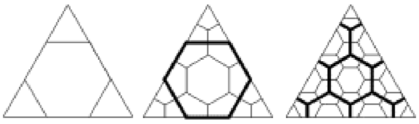


Figure 35: 1-to-3 hexagonal refinement applied to a triangle, as used to generate the ISEA3H. Image taken from [14].

However, due to the incongruent nature of hexagonal refinements, indexing the cells of DGGSs with hexagonal cells like the ISEA3H is not straightforward. PYXIS indexing is a hierarchy-based method designed for the cells of the ISEA3H, where the initial faces of the truncated icosahedron each receive an alphanumeric index [6, 98, 85] (see Figure 36).

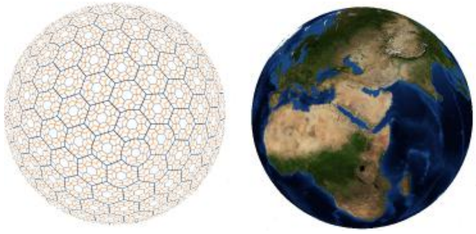


Figure 36: Hexagonal cells at three successive resolutions in PYXIS's system. Image taken from [6].

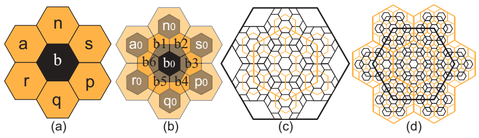


Figure 37: (a) Type *A* cells (orange) surround a type *B* cell (black) with index *b*. (b) The children of the cells illustrated in (a). (c), (d) The descendants of type *A* and *B* cells, respectively, after five successive refinements. Notice the fractal boundary developing at the finer resolutions.

In order to index cells at higher resolutions, the cells are categorized into two types: *A* cells and *B* cells. A fine cell that shares a centroid with a coarse cell is taken to be type *B* and is assigned as a child of that coarse cell. Fine cells that do not share a centroid with a coarse cell are labeled type *A* and are assigned as children to the nearest (coarse) type *B* cell. Hence, a type *A* cell has one type *B* parent and one type *B* child, whereas a type *B* cell can have a type *A* or *B* parent and has seven children: one type *B* cell surrounded by six type *A* cells. As shown in Figure 37, each cell's descendants at a fine resolution form fractal shape boundaries that fit together and cover the entire spherical icosahedron.

A similar hierarchy-based approach was proposed by Sahr [14], 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.

CPI: Sahr later proposed Central Place Indexing (CPI), a hierarchy-based indexing scheme designed for hexagonal cells that are refined by aligned 1-to-7, 1-to-4, or 1-to-3 refinements [87]. 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.

OA3HDGG and **OA4HDGG**: 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 [84]. This indexing assigns to the vertices the coordinates $(\pm 1, 0, 0)$, $(0, \pm 1, 0)$, and $(0, 0, \pm 1)$. 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 [90]. Snyder's equal area projection [95] can be used in both cases to obtain equal area cells on the sphere.

HQBS: 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 [153, 86]. 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 38).

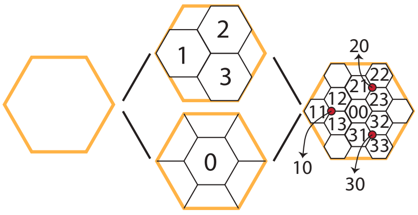


Figure 38: Combining two types of 1-to-4 refinement leads to a hierarchy-based indexing method for hexagons.

CDB: A latitude/longitude tiling of the Earth (resulting in quad cells) based on WGS-84 is employed by Presagis's Common Database (CDB) API [154]. 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, as described in Table 1. Sixteen levels of resolution are generated via 1-to-4 refinement. The CDB API has been released and is available for public use for data integration, simulation, and data exchange [155].

Table 1: The five zones defined in the CDB. Lat: latitude, Long: longitude.

Zone	Lat-Range	Lat-Interval	Long-Interval
I	0-50 N-S	1	1
II	50-70 N-S	1	2
III	70-75 N-S	1	3
IV	75-80 N-S	1	4
V	80-90 N-S	1	6

C-squares: The Concise Spatial Query and Representation System, or C-squares for short, also discretizes the Earth using lat/long coordinates [156]. The underlying coordinate domain is divided into four segments (NE, SE, NW, SW). Each cell then receives an index of the form *iyxx*; 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, a multiresolution hierarchy is not provided. It was invented and developed by CSIRO Marine and Atmospheric Research [157] for the purposes of mapping, spatial search, and environmental monitoring [158].

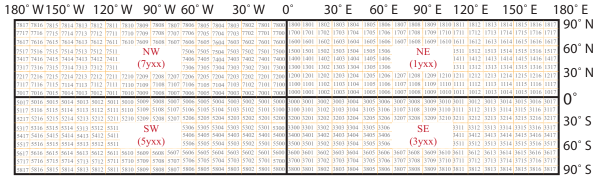


Figure 39: An example of indexing in the C-squares system.

Google Earth: As noted in [28, 159, 126], Google Earth uses a simple cylindrical projection with Clip-Maps, also known as Universal Textures, which were proposed by Tanner et al. in [143] and patented in [160]. Clip-Maps (see Figure 40) are partial mip-maps whose dimensions can be clipped to a maximum "clip size", altering the pyramidal shape of the mip-maps into one more closely resembling an obelisk. 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, and can be extended to ellipsoids as in the work of [125] or geometry as in the work of [144].

This type of image-based representation of the Earth with an underlying 2D domain has also been used in Bing Maps (which employs a hierarchy-based indexing method based on quadtrees [131] as shown in Figure 41), Skyline Globe, and Nasa World Wind [161, 162, 163].

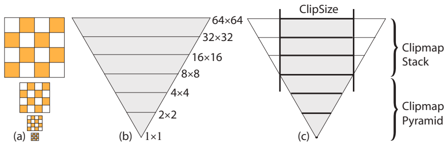


Figure 40: (a) An image at several different resolutions. (b) These images produce a pyramidal mip-map structure. (c) Using a ClipSize, a mip-map can be converted to a Clip-Map.

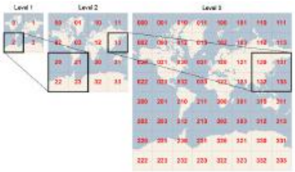


Figure 41: Bing Maps indexing mechanism at three successive resolutions. Image taken from [161].

Google Maps: Much like Bing Maps, Google Maps is not a true Digital Earth application due to the lack of a 3D multiresolution representation of the Earth, but employs similar techniques to those used in DGGSs. Unlike Google Earth, Google Maps uses a Mercator projection to flatten the Earth onto a 2D map [129]. The coarsest map resolution in Google Maps is an image tile with 256 × 256 cells, with finer resolutions obtained by dividing the tile into four new 256 × 256 tiles via 1-to-4 refinement [129]. The 2D coordinate-based indexing method on each tile takes index $(0, 0)$ to be located at the northwest corner (Figure 42) [129], with x values (longitude) increasing to the east and y values (longitude) increasing to the south. A visualization of this indexing scheme is provided by Maptiler [164].

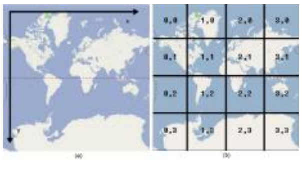


Figure 42: (a) Google Maps lat/long-based coordinate system. (b) The indices of each tile. Image taken from [129].

Cesium and **GlobWeb**: Mercator projection has also been used in Cesium [138, 165], a cross-platform and cross-browser web representation of the Earth that employs WebGL. To establish a hierarchy between cells, a 1-to-4 refinement is used and the UV parametrization of the 2D domain is used to define 2D coordinate-based cell indices. Similarly, GlobWeb has also used WebGL to implement a virtual Earth [166].

SDOG: While Digital Earth systems are typically concerned only with what is present on the surface of the Earth, it is possible to construct volumetric DGGSs that can store data for locations both above or below the Earth's surface. The Spheroid Degenerated-Octree Grid (SDOG), for instance, 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) [167]. 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 triangles near the poles; see Figure 43).

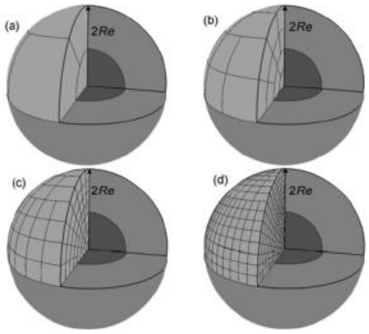


Figure 43: The SDOG system at three successive resolutions. Triangles at the poles make the octree degenerate. Image taken from [168].

Indexing of the grids is performed using two methods: Single Hierarchical Degenerated Z-curve Filling (SDZ), and Multiple Hierarchical Degenerated Z-curve Filling [168]. Both methods are based on a modified Z-curve defined on a congruent, unaligned 1-to-4 refinement. The SDZ scheme indexes an octree in base 10 (Figure 44), whereas the MDZ scheme provides a base 8 hierarchical indexing (Figure 45).

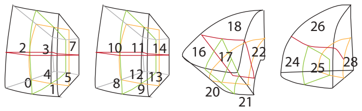


Figure 44: SDZ indexing uses a decimal base to index cells within a single resolution.

Whether or not a volumetric DGGS is necessary depends strongly on the intended application of the Digital Earth system, of which there are many.

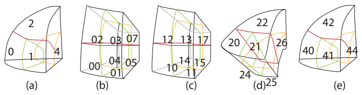


Figure 45: MDZ indexing. (a) Indices of cells at the first resolution. (b), (c), (d), (e) Indices of, respectively, the children of cells 0, 1, 2, and 4.

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 [169, 170, 72, 171, 172, 173, 174, 172, 175, 176, 177]. 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 [178], automatic quality control for 3D reconstructions [179], sharing geological data sets through a common framework [180, 181], and navigation and traffic simulation [182, 183, 184] are just some of the applications for which the Digital Earth has been employed.

Aside from these specific applications, there are a number of general areas in which both GIS and Digital Earth systems excel and are regularly used. In the following section, we discuss particular applications in which Digital Earths have been extensively employed.

5.1. Environmental Applications

One of the main functions of the Digital Earth is environmental data 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 [70, 185, 186, 187, 188].

5.1.1. Environment Monitoring

Observing, visualizing and monitoring environmental data 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 CO_2 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 [189]. 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 [190, 191, 192, 193, 194], 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 [195, 196], cropland area estimation [197], and visualization of geochemical rock and sediment data sets [198].

5.1.2. Disaster Prediction

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 [199]. For instance, although modeling and visualizing tsunamis is generally hard, the Digital Earth has proven helpful [200, 201] due to the relative ease of visualizing and integrating different data sets. The same can be said for earthquake visualization [202], real-time fire alert system design [203], and forecasting of flash flooding [204].

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 simple and generally accurate means through which animal [205] and plant [206, 207, 208] diversity and density may be monitored and landscapes that are not easily accessible may be studied [209]. Google Earth has been used in this area to study and monitor species habitats [210, 211] and marine life [212, 213, 214].

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. Using Digital Earth systems, we can integrate, visualize and manage disease-related data sets in order to analyze the causes and spread of diseases and their resistance to drugs [215, 216]. Using Google Earth, a low-cost surveillance system for Dengue fever was developed in [217], for the spread of avian influenza (H5N1) in [218], and for injury data collection in low income countries in [219]. In addition, Zhang, Shi, and Zhang used Google Maps to recognize patterns in data sets featuring epidemiological and geographical information [220].

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 [221], the construction of which is an area of extensive study in computer graphics [66].

Owing to the relative ease of integrating and visualizing geospatial data in a Digital Earth and its usefulness in the distribution, access, manipulation, and presentation of massive location-based data sets [222], the Digital Earth provides a handy infrastructure for the modeling of 3D cities. In [223], 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 (see Figure 46), and in [224] is used to support public participation in urban planning projects.

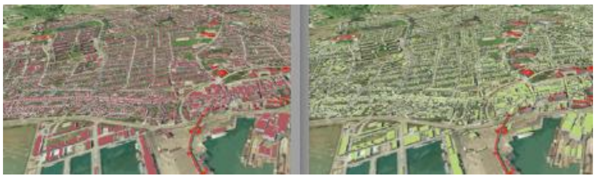


Figure 46: Houses with high electricity usage are highlighted in red. Image taken from [223].

5.4. Education

Beyond facilitating data 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 [225]. An interactive 3D environment that displays the Earth rather than a 2D map conveys a better sense of geographical information and spatial distances and better engages students. A survey of this framework as applied to teaching applications can be found in [226], 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 [227], 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 [228] 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 [229], 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 [230, 231, 232, 233].

In education and the applications discussed above, the Digital Earth has been successfully employed and has proven its effectiveness. Hence, the possibility for further use of Digital Earth systems in these application areas, and those yet to be explored, is ripe for further investigation.

6. Conclusion and Future Work

Although there exists an impressive body of work related to the Digital Earth and DGGSs, many research questions and challenges both fundamental and practical remain unsolved. These challenges primarily concern big data representation, creative visualization for geospatial data sets, and the handling of dynamic and unpredictable data sets such environmental data, among others. As these challenges are addressed, the usefulness of the Digital Earth framework as a reference model for the integration, analysis, and visualization of geospatial data can only increase.

In this survey, we have provided an overview on the construction of Digital Earths and the state-of-the-art in the area for the benefit of interested researchers and engineers. It is our hope that both the computer graphics and GIS communities will find this survey to be both useful and inspirational, and will pave the way for future collaborations.

Acknowledgments

We would like to thank anonymous reviewers for their thoughtful comments in the review process. We also thank Perry Peterson, Idan Shatz, and Nader Hamekasi for their insightful discussions and comments. This research was supported in part by the National Science and Engineering Research Council of Canada and PYXIS Innovation Inc.

References

[1] T. Funkhouser, Five principles for choosing research problems in computer graphics, ACM SIGGRAPH 2014 Computer Graphics Achievement Award Talk (2014).

[2] J. Chen, X. Zhao, Z. Li, An algorithm for the generation of Voronoi diagrams on the sphere based on QTM, Photogrammetric Engineering & Remote Sensing 69 (1) (2004) 79–89.

[3] H. Lukatela, A seamless global terrain model in the Hipparchus system, in: Discrete Global Grids: A Web Book, University of California, Santa Barbara, 2002, http://www. ncgia. ucsb. edu/globalgrids-book.

[4] D. R. Brooks, Grid systems for Earth radiation budget experiment applications, NASA Technical Memorandum, NASA Scientific and Technical Information Branch, 1981.

[5] Google Inc., Google Earth, `http://earth.google.com`.

[6] PYXIS innovation Inc., How PYXIS works, `http://www.pyxisinnovation.com/pyxwiki/index.php?title=How_PYXIS_Works` (2015).

[7] Virtual Terrain Project, `http://vterrain.org/` (2015).

[8] A. Gruen, GEO Informatics — In the spotlight — Satellite versus aerial images — not always a matter of choice!, `http://www.geoinformatics.com/blog/in-the-spotlight/satellite-versus-aerial-images-not-always-a-matter-of-choice` (2012).

[9] J. Zhang, P. Atkinson, M. F. Goodchild, Scale in Spatial Information and Analysis, CRC Press, 2014.

[10] L. Williams, Pyramidal parametrics, ACM SIGGRAPH Computer Graphics 17 (3) (1983) 1–11.

[11] C. Han, H. Hoppe, Optimizing continuity in multiscale imagery, ACM Transactions on Graphics 29 (6) (2010) 171:1–171:10.

[12] M. Lancelle, D. W. Fellner, Smooth transitions for large scale changes in multi-resolution images, in: Vision, Modeling, and Visualization, The Eurographics Association, 2011.

[13] R. Kooima, J. Leigh, A. Johnson, D. Roberts, M. SubbaRao, T. A. DeFanti, Planetary-scale terrain composition, IEEE Transactions on Visualization and Computer Graphics 15 (5) (2009) 719–733.

[14] K. Sahr, Location coding on icosahedral aperture 3 hexagon discrete global grids, Computers, Environment and Urban Systems 32 (3) (2008) 174–187.

[15] J. B. Campbell, R. H. Wynne, Introduction to Remote Sensing, 5th Edition, Guilford Publications, 2011.

[16] E. Mikhail, J. Bethel, J. McGlone, Introduction to Modern Photogrammetry, Wiley, 2001.

[17] H. Zhou, J. Sun, G. Turk, J. M. Rehg, Terrain synthesis from digital elevation models, IEEE Transactions on Visualization and Computer Graphics 13 (4) (2007) 834–848.

[18] O. Šťava, B. Beneš, M. Brisbin, J. Křivánek, Interactive terrain modeling using hydraulic erosion, in: Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '08, Eurographics Association, 2008, pp. 201–210.

[19] F. P. Tasse, A. Emilien, M.-P. Cani, S. Hahmann, A. Bernhardt, First person sketch-based terrain editing, in: Proc. of the Graphics Interface Conference, GI '14, Canadian Information Processing Society, 2014, pp. 217–224.

[20] J. Schneider, T. Boldte, R. Westermann, Real-time editing, synthesis, and rendering of infinite landscapes on GPUs, in: Proc. of Vision, Modeling and Visualization, 2006.

[21] D. S. Ebert, Texturing & Modeling: A Procedural Approach, Morgan Kaufmann, 2003.

[22] J. Brosz, F. F. Samavati, M. C. Sousa, Terrain synthesis by-example, Communications in Computer and Information Science: Advances in Computer Graphics and Computer Vision 4 (2008) 58–77.

[23] E. Danovaro, L. De Floriani, P. Magillo, E. Puppo, D. Sobrero, Level-of-detail for data analysis and exploration: A historical overview and some new perspectives, Computers & Graphics 30 (3) (2006) 334–344.

[24] L. De Floriani, E. Puppo, A hierarchical triangle-based model for terrain description, in: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1992, pp. 236–251.

[25] R. Pajarola, E. Gobbetti, Survey of semi-regular multiresolution models for interactive terrain rendering, The Visual Computer 23 (8) (2007) 583–605.

[26] K. Weiss, L. De Floriani, Simplex and diamond hierarchies: Models and applications, Computer Graphics Forum 30 (8) (2011) 2127–2155.

[27] M. Natali, E. M. Lidal, J. Parulek, I. Viola, D. Patel, Modeling terrains and subsurface geology, in: EuroGraphics 2013 State of the Art Reports (STARs), 2013, pp. 155–173.

[28] P. Cozzi, K. Ring, 3D Engine Design for Virtual Globes, 1st Edition, CRC Press, 2011.

[29] A. Mahdavi-Amiri, E. Harrison, F. F. Samavati, Hexagonal connectivity maps for digital Earth, International Journal of Digital Earth (2014) 1–20.

[30] P. Longley, Geographic Information Systems and Science, 2nd Edition, Wiley, 2005.

[31] A. Boyko, T. Funkhouser, Extracting roads from dense point clouds in large scale urban environment, ISPRS Journal of Photogrammetry and Remote Sensing 66 (6) (2011) S2–S12.

[32] F. Rottensteiner, Automatic generation of high-quality building models from lidar data, IEEE Computer Graphics and Applications 23 (6) (2003) 42–50.

[33] J. B. Mena, State of the art on automatic road extraction for GIS update: a novel classification, Pattern Recognition Letters 24 (16) (2003) 3037–3058.

[34] S. Clode, F. Rottensteiner, P. Kootsookos, E. Zelniker, Detection and vectorization of roads from lidar data, Photogrammetric Engineering & Remote Sensing 73 (5) (2007) 517–535.

[35] X. Hu, C. V. Tao, Y. Hu, Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35 (2) 88–92.

[36] C. Heipke, H. Mayer, C. Wiedemann, O. Jamet, Evaluation of automatic road extraction, International Archives of Photogrammetry and Remote Sensing 32 (3) 151–160.

[37] A. Fortier, D. Ziou, C. Armenakis, S. Wang, Survey of work on road extraction in aerial and satellite images, Tech. rep., Département de mathématiques et dínformatique, Université de Sherbrooke (1999).

[38] A. Shackelford, C. Davis, Fully automated road network extraction from high-resolution satellite multispectral imagery, in: Proc. of the IEEE Internation Geoscience and Remote Sensing Symposium, 2003, Vol. 1 of IGARSS '03, 2003, pp. 461–463.

[39] M.-F. A. Fortier, D. Ziou, C. Armenakis, S. Wang, Automated correction and updating of road databases from high-resolution imagery, Canadian Journal of Remote Sensing 27 (1) (2001) 76–89.

[40] H. Mayer, I. Laptev, A. Baumgartner, Multi-scale and snakes for automatic road extraction, in: Computer Vision — ECCV'98, Vol. 1407 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1998, pp. 720–733.

[41] M. A. Brovelli, M. Cannata, U. M. Longoni, Managing and processing LIDAR data within GRASS, in: Proc. of the

GRASS Users Conference, Vol. 29, 2002.

[42] J. P. McCrae, Sketch-based path design, Master's thesis, University of Toronto (2008).

[43] C. S. Applegate, S. D. Laycock, A. M. Day, A sketch-based system for highway design with user-specified regions of influence, Computers & Graphics 36 (6) (2012) 685–695, 2011 Joint Symposium on Computational Aesthetics (CAe), Non-Photorealistic Animation and Rendering (NPAR), and Sketch-Based Interfaces and Modeling (SBIM).

[44] G. Chen, G. Esch, P. Wonka, P. Müller, E. Zhang, Interactive procedural street modeling, ACM Transactions on Graphics 27 (3) (2008) 103:1–103:10.

[45] C. S. Applegate, S. D. Laycock, A. M. Day, A sketch-based system for highway design, in: Proc. of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, ACM, 2011, pp. 55–62.

[46] R. Smelik, T. Tutenel, K. J. de Kraker, R. Bidarra, Integrating procedural generation and manual editing of virtual worlds, in: Proc. of the Workshop on Procedural Content Generation in Games, ACM, 2010, pp. 2:1–2:8.

[47] E. Galin, A. Peytavie, N. Maréchal, E. Guérin, Procedural generation of roads, Computer Graphics Forum 29 (2) (2010) 429–438.

[48] R. Smelik, T. Tutenel, K. de Kraker, R. Bidarra, A declarative approach to procedural modeling of virtual worlds, Computers & Graphics 35 (2) (2011) 352 – 363.

[49] A. Emilien, A. Bernhardt, A. Peytavie, M.-P. Cani, E. Galin, Procedural generation of villages on arbitrary terrains, The Visual Computer 28 (6-8) (2012) 809–818.

[50] R. M. Smelik, T. Tutenel, R. Bidarra, B. Benes, A survey on procedural modelling for virtual worlds, Computer Graphics Forum 33 (6) (2014) 31–50.

[51] M. Schneider, R. Klein, Efficient and accurate rendering of vector data on virtual landscapes, in: Proc. of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG '07, Václav Skala-UNION Agency, 2007.

[52] M. Schneider, M. Guthe, R. Klein, Real-time rendering of complex vector data on 3D terrain models, in: Proc. of the 11th International Conference on Virtual Systems and Multimedia, 2005, pp. 573–582.

[53] M. Vaaraniemi, M. Treib, R. Westermann, High-quality cartographic roads on high-resolution DEMs, in: Proc. of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG '07, Václav Skala-UNION Agency, 2011.

[54] B. Deng, D. Xu, J. Zhang, C. Song, Visualization of vector data on global scale terrain, in: Proc. of the 2nd International Conference on Computer Science and Electronics Engineering, Atlantis Press, 2013.

[55] M. Zhou, J. Chen, J. Gong, A virtual globe-based vector data model: quaternary quadrangle vector tile model, International Journal of Digital Earth (ahead-of-print) (2015) 1–22.

[56] O. Kersting, J. Döllner, Interactive 3D visualization of vector data in GIS, in: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, ACM, 2002, pp. 107–112.

[57] A. Schilling, J. Basanow, A. Zipf, Vector based mapping of polygons on irregular terrain meshes for web 3D map services., in: Proc. of the 3rd International Conference on Web Information Systems and Technologies, WEBIST '07, 2007, pp. 198–205.

[58] A. Agrawal, M. Radhakrishna, R. Joshi, Geometry-based mapping and rendering of vector data over LOD phototextured 3D terrain models, in: Proc. of WSCG '06, 2006.

[59] Z. Qiao, J. Weng, Z. Sui, H. Cai, X. Zhang, A rapid visualization method of vector data over 3D terrain, in: Proc. of the 19th International Conference on Geoinformatics, 2011, pp. 1–5.

[60] S. Wenbin, S. Shigang, C. Feng, Z. Lichao, Geometry-based mapping of vector data and DEM based on hierarchical longitude/latitude grids, in: Proc. of the 2nd IITA International Conference on Geoscience and Remote Sensing, Vol. 1 of IITA-GRS '10, 2010, pp. 215–218.

[61] Z. Wartell, E. Kang, T. Wasilewski, W. Ribarsky, N. Faust, Rendering vector data over global, multi-resolution 3D terrain, in: Proc. of the Symposium on Data Visualisation, VISSYM '03, 2003, pp. 213–222.

[62] S. Nebiker, S. Bleisch, M. Christen, Rich point clouds in virtual globes – a new paradigm in city modeling?, Computers, Environment and Urban Systems 34 (6) (2010) 508–517.

[63] J. Otepka, S. Ghuffar, C. Waldhauser, R. Hochreiter, N. Pfeifer, Georeferenced point clouds: A survey of features and point cloud management, ISPRS International Journal of Geo-Information 2 (4) (2013) 1038–1065.

[64] A. Hengel, A. Dick, T. Thormahlen, B. Ward, P. H. S. Torr, Interactive 3D model completion, in: Proc. of Digital Image Computing: Techniques and Applications, DICTA '07, IEEE, 2007, pp. 175–181.

[65] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, R. Szeliski, Building Rome in a day, Communications of the ACM 54 (10) (2011) 105–112.

[66] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, W. Purgathofer, A survey of urban reconstruction, Computer Graphics Forum 32 (6) (2013) 146–177.

[67] C. Zhang, Fundamentals of Environmental Sampling and Analysis, Wiley, 2007.

[68] Food and Agriculture Organization of the United Nations, Guidelines for the Routine Collection of Capture Fishery Data, Vol. 382 of FAO Fisheries Technical Paper, 1999.
URL https://books.google.ca/books?id=hf71w8WYGRIC

[69] J. Vesely, Guidance for Choosing a Sampling Design for Environmental Data Collection, U. S. Environmental Protection Agency, Proquest, Umi Dissertation Publishing, 2011.

[70] D. White, J. A. Kimerling, S. W. Overton, Cartographic and geometric components of a global sampling design for environmental monitoring, Cartography and Geographic Information Science 19 (1) (1992) 5–22.

[71] M. F. Goodchild, Discrete global grids for digital Earth, in: Proc. of the 1st International Conference on Discrete Global Grids, 2000.

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

[73] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, R. Scopigno, Planet-sized batched dynamic adaptive meshes (P-BDAM), in: Proc. of IEEE Visualization, VIS '03, IEEE Computer Society, 2003, pp. 147–155.

[74] N. Greene, Environment mapping and other applications of world projections, IEEE Computer Graphics and Applications 6 (11) (1986) 21–29.

[75] K. Compton, J. Grieve, E. Goldman, O. Quigley, C. Stratton, E. Todd, A. Willmott, Creating spherical worlds, in: SIGGRAPH '07: ACM SIGGRAPH 2007 sketches, ACM, 2007.

[76] C. M. Grimm, Simple manifolds for surface modeling and parameterization, in: Proc. of Shape Modeling International, SMI '02, 2002, pp. 237–244.

[77] H. Alborzi, Geometric issues in spatial indexing, Master's thesis, University of Maryland, College Park (2006).

[78] G. H. Dutton, A Hierarchical Coordinate System for Geo-

processing and Cartography, Lecture Notes in Earth Sciences, Springer Berlin Heidelberg, 1999.

[79] M. Lee, H. Samet, Traversing the triangle elements of an icosahedral spherical representation in constant time, in: Proc. of the 8th International Symposium on Spatial Data Handling, 1998, pp. 22–33.

[80] K. Sahr, Hexagonal discrete global grid systems for geospatial computing, Archives of Photogrammetry, Cartography and Remote Sensing 22 (2011) 363–376.

[81] X. He, W. Jia, Hexagonal structure for intelligent vision, in: Proc. of the 1st International Conference of Information and Communication Technologies, ICIT '05, 2005, pp. 52–64.

[82] B. Kamgar-Parsi, B. Kamgar-Parsi, W. A. Sander III, Quantization error in spatial sampling: comparison between square and hexagonal pixels, in: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '89, 1989, pp. 604–611.

[83] J. Snyder, D. Mitchell, Sampling-efficient mapping of spherical images, Tech. rep., Microsoft Research (2001).

[84] A. Vince, Indexing the aperture 3 hexagonal discrete global grid, Journal of Visual Communication and Image Represention 17 (6) (2006) 1227–1236.

[85] A. Vince, X. Zheng, Arithmetic and Fourier transform for the PYXIS multi-resolution digital Earth model, International Journal of Digital Earth 2 (1) (2009) 59–79.

[86] X. Tong, J. Ben, Y. Wang, Y. Zhang, T. Pei, Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system, International Journal of Geographical Information Science 27 (5) (2013) 898–921.

[87] K. Sahr, Central place indexing systems, US Patent Application 20120206494 (filed October 28, 2010) (2010).

[88] D. White, Global grids from recursive diamond subdivisions of the surface of an octahedron or icosahedron, Environmental Monitoring and Assessment 64 (1) (2000) 93–103.

[89] M. F. Goodchild, Y. Shiren, A hierarchical spatial data structure for global geographic information systems, CVGIP: Graphical Models and Image Processing 54 (1) (1992) 31–44.

[90] J. Ben, X. Tong, R. Chen, A spatial indexing method for the hexagon discrete global grid system, in: Proc. of the 18th International Conference on Geoinformatics, 2010, pp. 1–5.

[91] H. Alborzi, H. Samet, Augmenting SAND with a spherical data model, in: Proc. of the International Conference on Discrete Global Grids,, 2000.

[92] SEEGrid, WebHome ¡ SCENZGrid ¡ SEEGrid, https://www.seegrid.csiro.au/wiki/SCENZGrid/WebHome.

[93] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, M. Bartelmann, HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere, The Astrophysical Journal 622 (2) (2005) 759.

[94] A. Mahdavi-Amiri, F. Bhojani, F. F. Samavati, One-to-two digital Earth, in: Proc. of the International Symposium on Visual Computing, ISVC '13, 2013, pp. 681–692.

[95] J. P. Snyder, An equal area map projection for polyhedral globes, Cartographica 29 (1992) 10–21.

[96] F. E. Wickman, E. Elvers, K. Edvarson, A system of domains for global sampling problems, Geografiska Annaler. Series A, Physical Geography 56 (3/4) (1974) 201–212.

[97] G. Fekete, L. A. Treinish, Sphere quadtrees: a new data structure to support the visualization of spherically distributed data, in: Proc. of SPIE 1259, Extracting Meaning from Complex Data: Processing, Display, Interaction, 1990, pp. 242–253.

[98] P. Peterson, Close-packed, uniformly adjacent, multiresolutional, overlapping spatial data ordering, US Patent 8,400,451 (issued March 19, 2013) (2004).

[99] A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunszt, P. Kukol, A. Thakar, Indexing the sphere with the hierarchical triangular mesh, Tech. Rep. MSR-TR-2005-123, Microsoft Research (2005).

[100] L. Middleton, J. Sivaswamy, Hexagonal Image Processing: A Practical Approach, Advances in Computer Vision and Pattern Recognition, Springer-Verlag London, 2005.

[101] I. P. Ivrissimtzis, N. A. Dodgson, M. A. Sabin, A generative classification of mesh refinement rules with lattice transformations, Computer-Aided Geometric Design 21 (1) (2004) 99–109.

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

[103] J. Thuburn, A PV-based shallow-water model on a hexagonal-icosahedral grid, Monthly Weather Review 125 (9) (1997) 2328–2347.

[104] X. Tong, J. Ben, Y. Wang, A new effective hexagonal discrete global grid system: Hexagonal quad balanced structure, in: Proc. of the 18th International Conference on Geoinformatics, 2010, pp. 1 –6.

[105] J. P. Snyder, Map Projections — A Working Manual, U.S. Government Printing Office, 1987.

[106] J. P. Snyder, Flattening the Earth: Two Thousand Years of Map Projections, University of Chicago Press, 1997.

[107] E. Grafarend, F. Krumm, Map projections: cartographic information systems, Springer, 2006.

[108] E. Harrison, Equal area spherical subdivision, Master's thesis (2012).

[109] F. Pearson, Map Projections: Theory and Applications, Taylor & Francis, 1990.

[110] K. Hormann, B. Lévy, A. Sheffer, Mesh parameterization: theory and practice, in: SIGGRAPH '07: ACM SIGGRAPH 2007 courses, ACM, 2007.

[111] M. Abate, F. Tovena, Curves and Surfaces, Springer Milan, 2012.

[112] M. P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, Inc., 1976, Ch. 4-6.

[113] R. Schmidt, C. Grimm, B. Wyvill, Interactive decal compositing with discrete exponential maps, ACM Transactions on Graphics 25 (3) (2006) 605–613.

[114] P. Grohs, J. Wallner, Definability and stability of multiscale decompositions for manifold-valued data, Journal of the Franklin Institute 349 (5) (2012) 1648–1664.

[115] S. R. Buss, J. P. Fillmore, Spherical averages and applications to spherical splines and interpolation, ACM Transactions on Graphics 20 (2) (2001) 95–126.

[116] D. Roşca, G. Plonka, Uniform spherical grids via equal area projection from the cube to the sphere, Journal of Computational Applied Mathematics 236 (6) (2011) 1033–1041.

[117] D. van Leeuwen, D. Strebe, A "slice-and-dice" approach to area equivalence in polyhedral map projections, Cartography and Geographic Information Science 33 (4) (2006) 269–286.

[118] A. Holhos, D. Roşca, An octahedral equal area partition of the sphere and near optimal configurations of points, Computers & Mathematics with Applications 67 (5) (2014) 1092–1107.

[119] Life presents R. Buckminster Fuller's Dymaxion world, LIFE Magazine.

[120] E. Praun, H. Hoppe, Spherical parametrization and remeshing, ACM Transactions on Graphics 22 (3) (2003) 340–349.

[121] A. Sheffer, C. Gotsman, N. Dyn, Robust spherical parameterization of triangular meshes, Springer, 2004.

[122] E. Harrison, A. Mahdavi-Amiri, F. F. Samavati, Optimization of inverse Snyder polyhedral projection, in: Proc. of the 2011

International Conference on Cyberworlds, CW '11, 2011, pp. 136–143.

[123] E. Harrison, A. Mahdavi-Amiri, F. F. Samavati, Analysis of inverse Snyder optimizations, Transactions on Computational Science 16 (2012) 134–148.

[124] M. Lambers, A. Kolb, Ellipsoidal cube maps for accurate rendering of planetary-scale terrain data, in: Proc. of the Pacific Conference on Computer Graphics and Applications, PG '12, 2012, pp. 5–10.

[125] A. M. Dimitrijević, D. D. Rančić, Ellipsoidal clipmaps — a planet-sized terrain rendering algorithm, Computers & Graphics 52 (2015) 43–61.

[126] Google Inc., Google Earth projection, https://support.google.com/earth/answer/148110?hl=en (2015).

[127] E. D. Kaplan, C. J. Hegarty, Understanding GPS: Principles and Applications, 2nd Edition, Artech House, Inc, 2006.

[128] National Geospatial-Intelligence Agency (NGA), Nga standardization document — implementation practice web mercator projection, http://earth-info.nga.mil/GandG/wgs84/web_mercator/%28U%29%20NGA_SIG_0011_1.0.0_WEBMERC.pdf.

[129] Google Inc., Google Maps coordinates, https://developers.google.com/maps/documentation/android/tileoverlay (2015).

[130] A. Mahdavi-Amiri, F. F. Samavati, P. Peterson, Categorization and conversions for indexing methods of discrete global grid systems, ISPRS International Journal of Geo-Information 4 (2015) 320–336.

[131] I. Gargantini, An effective way to represent quadtrees, Communications of the ACM 25 (12) (1982) 905–910.

[132] V. Pascucci, R. J. Frank, Global static indexing for real-time exploration of very large regular grids, in: Proc. of the ACM/IEEE Conference on Supercomputing, IEEE, 2001, pp. 45–45.

[133] J. Bai, X. Zhao, J. Chen, Indexing of the discrete global grid using linear quadtree, 2005, pp. 267–270.

[134] J. J. Bartholdi, III, P. Goldsman, Continuous indexing of hierarchical subdivisions of the globe, International Journal of Geographical Information Science 15 (2000) 489–522.

[135] A. Mahdavi-Amiri, F. F. Samavati, Connectivity maps for subdivision surfaces, in: GRAPP/IVAPP, 2012, pp. 26–37.

[136] A. Mahdavi-Amiri, F. F. Samavati, Atlas of connectivity maps, Computers & Graphics 39 (2014) 1 – 11.

[137] G. Sellers, J. Obert, P. Cozzi, K. Ring, E. Persson, J. de Vahl, J. M. P. van Waveren, Rendering massive virtual worlds, in: SIGGRAPH '13: ACM SIGGRAPH 2013 courses, ACM, 2013.

[138] P. Cozzi, D. Bagnell, A WebGL globe rendering pipeline, in: GPU Pro 4: Advanced Rendering Techniques, A. K. Peters/CRC Press, 2013.
URL https://books.google.se/books?id=TUuhiPLNmbAC

[139] T. Bernardin, E. Cowgill, O. Kreylos, C. Bowles, P. Gold, B. Hamann, L. Kellogg, Crusta: A new virtual globe for real-time visualization of sub-meter digital topography at planetary scales, Computers and Geosciences 37 (1) (2011) 75–85.

[140] R. Westerteiger, A. Gerndt, H. Hagen, B. Hamann, A rapid visualization method of vector data over 3D terrain, in: Proc. of the IRTG Workshop on Visualization of Large and Unstructured Data Sets, VLUDS '11, 2011, pp. 13–23.

[141] C. Thorne, Using a floating origin to improve fidelity and performance of large, distributed virtual worlds, in: Proc. of the 2005 International Conference on Cyberworlds, CW '05, 2005.

[142] D. Ohlarik, Precisions, precisions, http://blogs.agi.com/insight3d/index.php/2008/09/03/precisions-precisions/ (2008).

[143] C. C. Tanner, C. J. Migdal, M. T. Jones, The clipmap: A virtual mipmap, in: Proc. of SIGGRAPH '98, Annual Conference Series, 1998, pp. 151–158.

[144] F. Losasso, H. Hoppe, Geometry clipmaps: Terrain rendering using nested regular grids, ACM Transactions on Graphics 23 (3) (2004) 769–776.

[145] M. Clasen, H.-C. Hege, Terrain rendering using spherical clipmaps, in: Proc. of the Eurographics Conference on Visualization, EUROVIS '06, 2006.

[146] A. Mahdavi-Amiri, F. F. Samavati, Adaptive atlas of connectivity maps, in: Proc. of the 8th International Conference on Curves and Surfaces, Lecture Notes in Computer Science, Springer, 2014.

[147] H.-R. Pakdel, F. F. Samavati, Incremental adaptive loop subdivision, in: Computational Science and Its Applications, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 237–246.

[148] J. Keysers, Review of digital globes 2015, http://www.crcsi.com.au/assets/Resources/Globe-review-paper-March-2015.pdf (2015).

[149] E. Hivon, K. Gorski, M. Reinecke, HEALPix, data analysis, simulations and visualization on the sphere, http://sourceforge.net/projects/healpix/.

[150] Landcare Research, Landcare Research - Manaaki Whenua, http://www.landcareresearch.co.nz/home.

[151] GNS Science, Home - GNS Science, http://www.gns.cri.nz/.

[152] G. H. Dutton, Zenithial orthotriangular projection, in: Proc. of Auto-Carto, Vol. 10, 1991, pp. 77–95.

[153] PLA Information Engineering University, Hierarchical encoding, operation and indexing method of hexagonal grid with aperture of 4, CN Patent 102,281,075 (issued March 6, 2013) (2013).

[154] Presagis, Common Database (CDB) — Presagis, http://www.presagis.com/products_services/standards/cdb/.

[155] Presagis, Common Database (CDB) API — Presagis, http://www.presagis.com/products_services/products/modeling-simulation/free_tools/cdb_api/.

[156] T. Rees, "C-Squares", a new spatial indexing system and its applicability to the description of oceanographic datasets, Oceanography 16 (1) (2003) 11–19.

[157] CSIRO, CSIRO marine and atmospheric research, http://www.cmar.csiro.au/.

[158] CSIRO, C-squares home page, http://www.cmar.csiro.au/csquares/.

[159] A. Bar-Zeev, How Google Earth [really] works, http://www.realityprime.com/blog/2007/07/how-google-earth-really-works/ (2007).

[160] C. Tanner, Asynchronous multilevel texture pipeline, uS Patent 6,618,053 (2003).

[161] J. Schwartz, Bing Maps tile system, http://msdn.microsoft.com/en-us/library/bb259689.aspx.

[162] Skyline Software Systems Inc., Skyline Globe, http://www.skylinesoft.com/SkylineGlobe/corporate/home/index.aspx? (2015).

[163] NASA, World Wind Java SDK, http://worldwind.arc.nasa.gov/java/.

[164] K. P. Přidal, Tiles à la google maps: Coordinates, tile bounds and projection, http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/ (2008).

[165] Analytical Graphics, Inc. (AGI), Cesium, http://cesiumjs.org/ (2015).

[166] Telespazio, Globweb, http://demonstrator.

telespazio.com/GlobWeb/ (2015).

[167] J.-Q. Yu, L.-X. Wu, G.-J. Zi, Z.-Z. Guo, SDOG-based multiscale 3D modeling and visualization on global lithosphere, Science China Earth Sciences 55 (6) (2012) 1012–1020.

[168] Y. Jie-Qing, W. Li-Xin, Spatial subdivision and coding of a global three-dimensional grid: Spheroid degenerated-octree grid, in: Proc. of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS '09, 2009, pp. II–361–II–364.

[169] D. Butler, Virtual globes: The web-wide world, Nature 439 (7078) (2006) 776–778.

[170] B. T. Tuttle, S. Anderson, R. Huff, Virtual globes: An overview of their history, uses, and future challenges, Geography Compass 2 (5) (2008) 1478–1505.

[171] K. E. Grossner, K. C. Clarke, Is Google Earth, "digital Earth?": Defining a vision, in: Proc. of the 5th International Symposium on Digital Earth, 2007.

[172] M. Craglia, K. de Bie, D. Jackson, M. Pesaresi, G. Remetey-Fülöpp, C. Wang, A. Annoni, L. Bian, F. Campbell, M. Ehlers, et al., Digital Earth 2020: towards the vision for the next decade, International Journal of Digital Earth 5 (1) (2012) 4–21.

[173] M. F. Goodchild, H. Guo, A. Annoni, L. Bian, K. de Bie, F. Campbell, M. Craglia, M. Ehlers, J. van Genderen, D. Jackson, et al., Next-generation digital Earth, Proceedings of the National Academy of Sciences 109 (28) (2012) 11088–11094.

[174] D. Li, Y. Yao, Z. Shao, L. Wang, From digital Earth to smart Earth, Chinese Science Bulletin 59 (8) (2014) 722–733.

[175] A. Annoni, M. Craglia, M. Ehlers, Y. Georgiadou, A. Giacomelli, M. Konecny, N. Ostlaender, G. Remetey-Fülöpp, D. Rhind, P. Smits, et al., A European perspective on digital Earth, International Journal of Digital Earth 4 (4) (2011) 271–284.

[176] M. F. Goodchild, The use cases of digital Earth, International Journal of Digital Earth 1 (1) (2008) 31–42.

[177] M. F. Goodchild, The future of digital Earth, Annals of GIS 18 (2) (2012) 93–98.

[178] A. Beltran, C. Abargues, C. Granell, M. Núñez, L. Díaz, J. Huerta, A virtual globe tool for searching and visualizing geo-referenced media resources in social networks, Multimedia Tools and Applications 64 (1) (2013) 171–195.

[179] C. Zhong, Y. Zhu, M. Li, F. Hu, A study on automatic quality control for 3D city reconstruction, in: Proc. of SPIE 6753, Geoinformatics 2007: Geospatial Information Science, 2007.

[180] S. Tavani, P. Granado, A. Corradetti, M. Girundo, A. Iannace, P. Arbués, J. A. Muñoz, S. Mazzoli, Building a virtual outcrop, extracting geological information from it, and sharing the results in Google Earth via OpenPlot and Photoscan: An example from the Khaviz Anticline (Iran), Computers & Geosciences 63 (2014) 44–53.

[181] G. Walker Johnson, A. G. Gaylord, J. C. Franco, R. P. Cody, J. J. Brady, W. Manley, M. Dover, D. Garcia-Lavigne, R. Score, C. E. Tweedie, Development of the arctic research mapping application (ARMAP): Interoperability challenges and solutions, Computers & Geosciences. 37 (11) (2011) 1735–1742.

[182] J. Sewall, D. Wilkie, M. C. Lin, Interactive hybrid simulation of large-scale traffic, ACM Transactions on Graphics 30 (6) (2011) 135.

[183] D. Wilkie, J. Sewall, M. C. Lin, Transforming GIS data into functional road models for large-scale traffic simulation, IEEE Transactions on Visualization and Computer Graphics 18 (6) (2012) 890–901.

[184] W. Doyle, J. M. Pearce, Utilization of virtual globes for open source industrial symbiosis, Open Environmental Sciences 3 (2009) 88–96.

[185] D. Williamson, Integration of the barotropic vorticity equation on a spherical geodesic grid, Tellus A 20 (4).

[186] J. Li, Y. Jiang, C. Yang, Q. Huang, M. Rice, Visualizing 3D/4D environmental data using many-core graphics processing units (GPUs) and multi-core central processing units (CPUs), Computers & Geosciences 59 (2013) 78–89.

[187] J. Li, H. Wu, C. Yang, D. W. Wong, J. Xie, Visualizing dynamic geosciences phenomena using an octree-based view-dependent LOD strategy within virtual globes, Computers & Geosciences 37 (9) (2011) 1295–1302.

[188] K. Johansen, C. Roelfsema, S. Phinn, High spatial resolution remote sensing for environmental monitoring and management preface, Journal of Spatial Science 53 (1) (2008) 43–47.

[189] T. A. Erickson, A. M. Michalak, J. C. Lin, A data system for visualizing 4-D atmospheric CO2 models and data, OSGeo Journal 8 (2011) 37–47.

[190] S. R. J. Sheppard, Landscape visualisation and climate change: the potential for influencing perceptions and behaviour, Environmental Science & Policy 8 (6) (2005) 637 – 654, mitigation and Adaptation Strategies for Climate Change.

[191] L. M. Ballagh, M. A. Parsons, R. Swick, Visualising cryospheric images in a virtual environment: present challenges and future implications, Polar Record 43 (04) (2007) 305–310.

[192] S. Li, C. Xiong, Z. Ou, A web GIS for sea ice information and an ice service archive, Transactions in GIS 15 (2) (2011) 189–211.

[193] Representing scientific data sets in KML: Methods and challenges, Computers & Geosciences 37 (1).

[194] M. C. Serreze, M. M. Holland, J. Stroeve, Perspectives on the arctic's shrinking sea-ice cover, Science 315 (5818) (2007) 1533–1536.

[195] P. Jankowski, M.-H. Tsou, R. D. Wright, Applying internet geographic information system for water quality monitoring, Geography Compass 1 (6) (2007) 1315–1337.

[196] E. S. Bradley, D. A. Roberts, P. E. Dennison, R. O. Green, M. Eastwood, S. R. Lundeen, I. B. McCubbin, I. Leifer, Google Earth and Google Fusion Tables in support of time-critical collaboration: Mapping the deepwater horizon oil spill with the AVIRIS airborne spectrometer, Earth Science Informatics 4 (4) (2011) 169–179.

[197] L. Liang, P. Gong, Evaluation of global land cover maps for cropland area estimation in the conterminous United States, International Journal of Digital Earth 8.

[198] Y. Yamagishi, K. Suzuki, H. Tamura, H. Yanaka, S. Tsuboi, Visualization of geochemical data for rocks and sediments in Google Earth: Development of a data converter application for geochemical and isotopic data sets in database systems, Geochemistry, Geophysics, Geosystems 12 (3).

[199] A. Kawasaki, M. L. Berman, W. Guan, The growing role of web-based geospatial technology in disaster response and support, Disasters 37 (2) (2013) 201–221.

[200] H. Zhang, Y. Shi, D. A. Yuen, Z. Yan, X. Yuan, C. Zhang, Modeling and visualization of tsunamis, in: Earthquakes: Simulations, Sources and Tsunamis, Springer, 2008, pp. 475–496.

[201] X. Yuan, M. X. Nguyen, Y. C. Liu, D. A. Yuen, B. Chen, Y. Shi, Tsunami and earthquake visualization inspired by light interference, in: Proc. of IEEE Visualization, 2006.

[202] L. H. Kellogg, G. W. Bawden, T. Bernardin, M. Billen, E. Cowgill, B. Hamann, M. Jadamec, O. Kreylos, O. Staadt, D. Sumner, Interactive visualization to advance earthquake simulation, in: Earthquakes: Simulations, Sources and Tsunamis, Springer, 2008, pp. 621–633.

[203] D. K. Davies, H. F. Vosloo, S. S. Vannan, P. E. Frost, Near real-time fire alert system in South Africa: from desktop to mobile

service, in: Proc. of the 7th ACM Conference on Designing Interactive Systems, ACM, 2008, pp. 315–322.

[204] T. Brice, A. Foster, The application of virtual globe software in forecasting flash flooding of low water crossings, in: Proc. of the 25th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 2009.

[205] B. Weber, Polar bears counted using Google Earth images, The Star.

[206] Y. D. Bar-Ness, Conservation of Indian Heritage Trees, Vol. 17, Indian National Trust on Arts and Cultural Heritage, 2009.

[207] A. Papatheodoulou, K. Michael, T. Greenwood, P. Klonis, L. Taylor, I. Kiriakou, M. R. Proteriotis, L. Sergides, M. Ioannou, M. Viktora, et al., The use of satellite imagery for the identification and protection of important oak areas, in: Proc. of the 1st International Conference on Remote Sensing and Geoinformation of Environment, International Society for Optics and Photonics, 2013.

[208] L. Reifke, Harnessing the power of Google Earth for seagrass conservation in the Comoros Islands, Master's thesis, Oregon State University (2009).

[209] J. A. González-Delgado, A. M. Martínez-Graña, J. Civis, F. J. Sierro, J. L. Goy, C. J. Dabrio, F. Ruiz, M. L. González-Regalado, M. Abad, Virtual 3D tour of the Neogene palaeontological heritage of Huelva (Guadalquivir Basin, Spain), Environmental Earth Sciences 73 (8) (2014) 4609–4618.

[210] P. M. Benham, E. J. Beckman, S. G. DuBay, L. M. Flores, A. B. Johnson, M. J. Lelevier, C. J. Schmitt, N. A. Wright, C. C. Witt, Satellite imagery reveals new critical habitat for endangered bird species in the high Andes of Peru, Endangered Species Research (2011) 145–157.

[211] P. P. Olea, P. Mateo-Tomás, Assessing species habitat using Google Street View: a case study of cliff-nesting vultures, PloS one 8 (1).

[212] Y. Shiau, Y. Chen, K. Tseng, J. Cheng, S. Lin, S. Lo, H. Chou, A real-time high-resolution underwater ecological observation streaming system, in: Proc. of the International Society for Photogrammetry and Remote Sensing Technical Commission VIII Symposium, 2010, pp. 517–521.

[213] F. Sheth, J.-P. Aurambout, C. Pettit, Creating a marine habitat geoexploratorium using digital globe technologies, in: Proc. of the 7th International Symposium on Digital Earth, 2011.

[214] M. S. Merrifield, W. McClintock, C. Burt, E. Fox, P. Serpa, C. Steinback, M. Gleason, MarineMap: a web-based platform for collaborative marine protected area planning, Ocean & Coastal Management 74 (2013) 67–76.

[215] A. Brandusescu, R. Sieber, N. Schuurman, The use of geovisualization to public health, in the context of open source applications and digital Earths: an effective representation?, in: Proc. of Spatial Knowledge and Information, Vol. 2, 2011.

[216] A.-S. Stensgaard, C. F. Saarnak, J. Utzinger, P. Vounatsou, C. Simoonga, G. Mushinge, C. Rahbek, F. Møhlenberg, T. K. Kristensen, Virtual globes and geospatial health: the potential of new tools in the management and control of vector-borne diseases, Geospatial Health 3 (2) (2009) 127–141.

[217] A. Y. Chang, M. E. Parrales, J. Jimenez, M. E. Sobieszczyk, S. M. Hammer, D. J. Copenhaver, R. P. Kulkarni, Combining Google Earth and GIS mapping technologies in a dengue surveillance system for developing countries, International Journal of Health Geographics 8 (1) (2009) 49.

[218] D. Janies, A. W. Hill, R. Guralnick, F. Habib, E. Waltari, W. C. Wheeler, Genomic analysis and geographic visualization of the spread of avian influenza (H5N1), Systematic Biology 56 (2) (2007) 321–329.

[219] J. Cinnamon, N. Schuurman, Injury surveillance in low-resource settings using geospatial and social web technologies, International Journal of Health Geographics 9 (1) (2010) 25.

[220] J. Zhang, H. Shi, Y. Zhang, Self-organizing map methodology and Google Maps services for geographical epidemiology mapping, in: Proc. of Digital Image Computing: Techniques and Applications, DICTA '09, IEEE, 2009, pp. 229–235.

[221] W. Förstner, 3D-city models: automatic and semiautomatic acquisition methods, in: Proc. of Photogrammetric Week '99, Wichmann Verlag, 1999, pp. 291–303.

[222] D.-I. L. Ross, Virtual 3d city models in urban land management, Ph.D. thesis, Technische Universität Berlin (2010).

[223] W. Wen, E. Kjems, L. Bodum, J. Kolar, Dynamic features in a 3D city model as an energy system, in: Proc. of the ISPRS International Conference on 3D Geoinformation, 2010.

[224] H. Wu, Z. He, J. Gong, A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes, Computers, Environment and Urban Systems 34 (4) (2010) 291–298.

[225] R. B. Schultz, J. J. Kerski, T. C. Patterson, The use of virtual globes as a spatial teaching tool with suggestions for metadata standards, Journal of Geography 107 (1) (2008) 27–34.

[226] R. Rakshit, Y. Ogneva-Himmelberger, Application of virtual globes in education, Geography Compass 2 (6) (2008) 1995–2010.

[227] C. Langford, New geo-spatial software connects with kids, Canadian Geographic.

[228] B. Thankachan, T. Franklin, Impact of Google Earth on student learning, International Journal of Humanities and Social Science (3) (2013) 11–16.

[229] A. Demirci, A. Karaburun, H. Klar, Using Google Earth as an educational tool in secondary school geography lessons, International Research in Geographical and Environmental Education 22 (4) (2013) 277–290.

[230] A. M. Bodzin, D. Anastasio, V. Kulo, Designing Google Earth activities for learning Earth and environmental science, in: Teaching Science and Investigating Environmental Issues with Geospatial Technology, Springer Netherlands, 2014, pp. 213–232.

[231] R. Hennessy, T. Arnason, I. Ratinen, L. Rubensdotter, Google Earth geo-education resources: A transnational approach from Ireland, Iceland, Finland, and Norway, Geological Society of America Special Papers 492 (2012) 413–418.

[232] T. C. Patterson, Google Earth as a (not just) geography education tool, Journal of Geography 106 (4) (2007) 145–152.

[233] A. M. Bodzin, Integrating instructional technologies in a local watershed investigation with urban elementary learners, The Journal of Environmental Education 39 (2) (2008) 47–58.