Spatial Data on the Web Best Practices – Tiling geospatial data

This chapter highlights best practices and relevant international standards related to tiling geospatial data. First, the foundational concepts behind tiling geospatial data are introduced. The *OGC 2D Tile Matrix Set and Tileset Metadata* standard is then presented, standardizing one of the most common approach to accessing tiled map data. A brief history of the origins and use of the concepts from this standard in the *OGC Web Map Tile Service (WMTS)*, *OGC Web Map Service (WMS)*, *OGC GeoPackage*, and the *OSGeo Tile Map Service (TMS)* is then included. The particularities of tiling raster data such as maps, imagery and gridded coverages are discussed. The subject of tiled vector feature data (*vector tiles*) is also covered. Specifics of tiling point clouds, and other types of 3D data are explored. An overview of the OGC API family of standards allowing to efficiently access geospatial data as tiles follows. The topic of Discrete Global Grid Systems based on 2D TileMatrixSets is addressed as well. The possibility to use tiles as a mechanism to facilitate parallel and on-demand processing is explained. Finally, thoughts are given on how tiles can help make Web maps more accessible.

Conceptual tiling of 2D, 3D, nD geospatial data

Due to the constrained nature of computing resources (memory, bandwidth, processing power), accessing, visualizing and analyzing large amounts of geospatial data poses certain challenges. Datasets which cover a large spatial region, while also being high resolution, benefit immensely from "tiling" the data into smaller pieces of data which can be processed or visualized individually, guaranteeing a deterministic upper bound to the resource requirements. This determinism, which is key to ensuring responsive and stable web mapping applications, can be achieved by making the size of the spatial area for these data bundles inversely proportional to their resolution: the larger the spatial region spanned by the tile, the lower the resolution. Since a maximum number of these tiles can also be visualized on a physical display of a given size and resolution, tiles can ensure the deterministic performance of a visualization system from microscopic to global scales.

The Open Geospatial Consortium (OGC) *Core Tiling Conceptual and Logical Models for 2D Euclidean Space* Abstract Specification Topic 22 (OGC 19-014r3) defines the conceptual basis for two-dimensional tiling of geospatial data. From that specification, *"in mathematics, a tiling (tessellation) is a collection of subsets of the space being tiled, i.e. tiles that cover the space without gaps or overlaps"*, and a tile is *"geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected "piece" (topological disc) without "holes" or "lines"."*

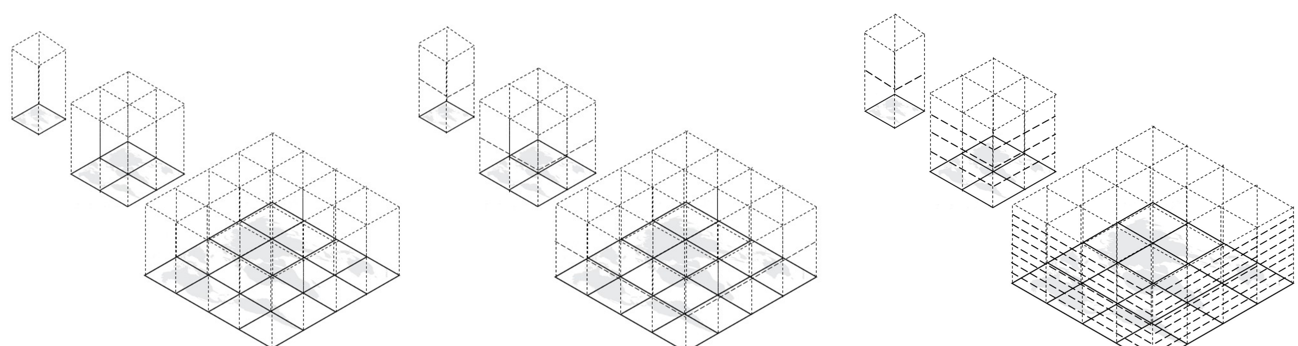
In practice and in OGC Implementation Standards, a tile can refer either to the *subset of the space being tiled*, or to *a bundle of data contained within such a tile*. The content carried by a tile can be of various data types including a visual representation ("map tiles"), vector data ("vector tiles"), values measured in a multi-dimensional grid ("gridded coverage tiles"), point clouds, or 3D models.

As defined in OGC 19-014r3, the space being tiled is in a particular coordinate reference system ("*coordinate system that is related to the real world by a datum"* – ISO 19111*),* where a coordinate system in turn is a "*set of mathematical rules for specifying how coordinates are to be assigned to points"* (ISO 19111). The foundation for coordinate reference system (CRS), and source of these definitions, is the OGC *Referencing by Coordinates* Abstract Specification Topic 2 (OGC 18-005r4) also known as ISO 19111. This facilitates integration of multiple geospatial datasets whether they are tiled or not. Different types of CRSs can be used for tiles including geographic CRS where a point specifies, for example, a WGS84 latitude and longitude, projected CRS where the Earth is projected to a 2D Cartesian space, and geocentric CRS where 3D Cartesian coordinates relative to the Earth center are used (in particular for 3D content).

To balance between the size of the spatial area with the resolution of the data contained within a tile, the CRS space can be tiled at different resolutions. A well-defined set of such tessellations for a particular CRS is called a *tiling scheme*. A set of tiles containing geospatial data tiled according to a particular tiling scheme is called a *tileset*. For client-server architectures, the use of a predefined tiling scheme presents an important advantage over on-demand requests for arbitrary spatial regions, allowing to cache data at multiple network nodes between a client and server, and also enables the possibility to host a large repository of pre-generated tiles on object storage as a more affordable option compared to the computing resources required to generate them on-demand.

There are multiple approaches to tile 3D geospatial data. The tiling may be a true tiling in the mathematical sense (a tessellation), or tiles of 3D content may be allowed to overlap, for example to allow the whole model of a building to be included in a tile. The tiling may be defined irrespective of the content, or may take into consideration the density of the data being tiled in an attempt to equalize the amount of data per tile. While it is possible to define tile boundaries using a projected CRS, a 3D geographic (e.g., latitude, longitude, and height above the WGS84 ellipsoid) or geocentric CRS (e.g., x, y, z Earth-centered Earth-fixed Cartesian coordinates) is typically used for 3D content, making that same CRS better suited for defining the 3D content of the tile as well.

A simple approach to define a regular 3D tiling scheme based on a geographic CRS is to extend a 2D tiling scheme vertically. The resulting columns can either be single three-dimensional tiles, or they can be subdivided at a regular linear or logarithmic interval. The number of divisions can remain the same or can increase at higher resolutions. The same extension approach can also be applied to other types of dimensions besides altitude, such as time and atmospheric pressure levels. These three different approaches to extending two-dimensional geospatial tiling to additional dimensions are described in more details in annex J of the *OGC 2D Tile Matrix Set and Tileset Metadata* Standard.



a) extension *b) extension and division* *c) octrees or orthtrees (hyperoctrees)*

Figure 1. The three approaches suggested in Annex J for extending a 2DTMS to additional dimensions

The OGC 2D Tile Matrix Set and Tileset Metadata **Standard**

The OGC *Two-Dimensional Tile Matrix Set and Tileset Metadata* Implementation Standard (2.0: [OGC 17-083r4]) defines one particular type of tiling scheme called a *2D Tile Matrix Set (2DTMS)*. A 2DTMS consists of one or more *tile matrix*, each tiling the CRS space at a given scale (zoom level). The tiling of a 2DTMS always follows a regular rectangular grid, meaning that the width and height in CRS units is the same for all tiles. When tiling a dataset according to a 2DTMS, which produces a tileset compatible with that 2DTMS, the resolution of the data within a tile (the number of cells units per CRS units across the tile) is also expected to be the same for all tiles of a particular tile matrix, often 256 x 256 pixels. Rectangular tiling allows for a very simple indexing mechanism to reference a specific tile of a tile matrix using two integral numbers: a row and a column. In 2DTMS version 2.0, rows can be counted either from the top or from the bottom corner, while columns are always counted from left to right. Tile matrices are referenced by an identifier which is not required to be a number, but using a consecutive sequence of integral numbers starting at 0 for the least detailed zoom level is a ***Best Practice*** used to define most common 2DTMSs. Together, the *tile matrix*, *tile row* and *tile column* uniquely identify a specific tile of a 2DTMS or of a tileset tiled according to that 2DTMS.

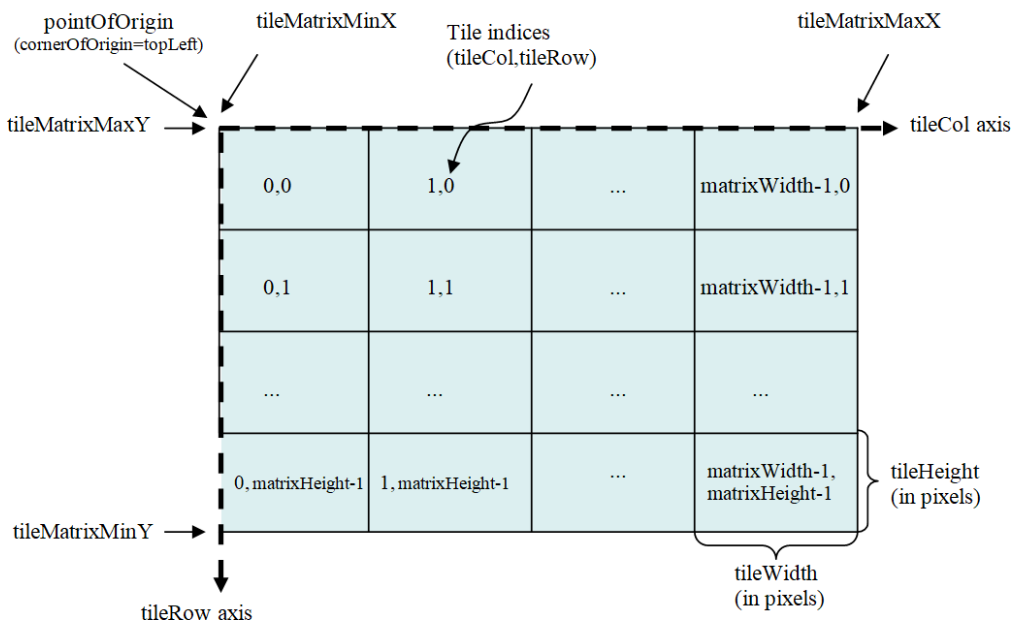


Figure 2. The main elements of a Tile Matrix Set definition, such as dimensions and tile indexing coordinates

Despite the zoom level being integral, sometimes a fractional number between two levels is used as a proxy for scale, in the context of a specific 2DTMS. Unless there is a particular need for defining tile matrices for specific scales, such as the *GlobalCRS84Pixel Well-known Scale Set* (WKSS) aiming to have rounded Degrees / Minutes / Seconds per pixels values, or the *GlobalCRS84Scale* WKSS selecting scales used in common cartographic products, it is a ***Best Practice*** to use a mathematical function to define the relationship from the scale of one tile matrix to the next, such as using a quad-tree (factor of two between two zoom levels).

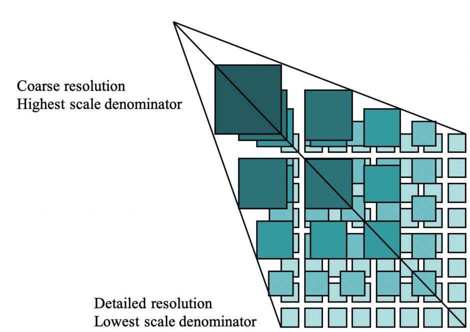


Figure 3. The tile pyramid concept of a 2DTMS

One exception to the grid being regular is made for *variable width tile matrices*, where contiguous tiles along a row can be coalesced into a single tile. This capability is particularly useful for geographic coordinate reference systems as it allows to define larger tiles in polar regions to reflect the smaller real-world physical distance between meridians and thus approach equal area tiles. Since all tiles feature the same data resolution of CRS units, this results in considerable space saving near the poles, and makes it possible to define a single global 2DTMS in a geographic CRS which is suitable for the entire world. The Morecantile and GDAL libraries include support for variable width tile matrices, and they form an important basis of the OGC CDB data store standard (OGC 15-113r6, 2.0: draft OGC 23-034) used extensively in the training and simulation domain, where deterministic performance is important and global high-resolution datasets are common.

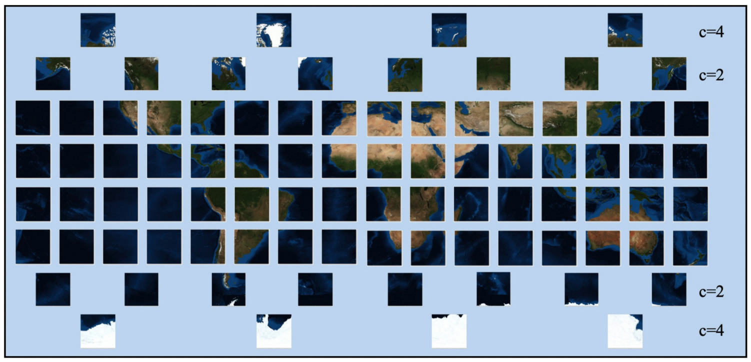


Figure 4. An example of a variable width Tile Matrix (level 2 of the *GNOSIS Global Grid*)

Although it is possible to define custom 2DTMSs for very specific needs, several 2DTMSs in various CRSs have wide-ranging applications. Registration with an authority assigning a specific URI to a particular definition facilitates reuse, recognition and interoperability for a common 2DTMS, similar to how EPSG codes have facilitated these characteristics for CRSs. The OGC maintains such a register of 2D Tile Matrix Sets. All 2DTMSs appearing in the 2DTMS informative annexes are included in this register, and additional definitions will likely be registered in the future. It is a ***Best Practice*** to re-use already registered 2DTMS, or if requiring the use of a particular grid that is not yet registered, to register such grid with the OGC, if it could be of notable use, for example in the case a grid commonly used by a particular national mapping authority. A new 2DTMS may be registered by submitting a pull request for the register to the OGC 2D Tile Matrix Set GitHub repository. Two encodings have been standardized for defining a 2DTMS: one using JSON and one using XML. The OGC register includes links to definitions in both encodings for each 2DTMS. Some visualization clients might only be able to display multiple tilesets together if they use the same tile matrix set, or if their 2DTMS are based on the same CRS.

The first part of the *OGC 2D Tile Matrix Set and Tileset Metadata* Standard, about *Tile Matrix Sets*, focuses on the concepts and encodings for defining a 2DTMS. The second part of the standard, about *Tileset Metadata*, provides a model and encoding for metadata describing a tileset tiled according to a specific 2DTMS. Three elements are required for any tileset metadata: the 2DTMS definition, the CRS for the tileset and the data type. The definition of the 2DTMS can be embedded directly within the tileset metadata for offline use cases, whereas it is required to be a link with an *[ogc-rel:tiling-scheme]* relation type for online use cases. The CRS of the tileset is usually the same as that of the 2DTMS definition, although the Standard specifies in the *6.2.1.1. TileMatrixSet CRS Compatibility* clause three cases where the tileset CRS may differ from that of the TileMatrixSet: the tileset CRS extending the 2DTMS with additional dimensions, the tileset CRS differing only by axis order (the 2DTMS axis order only affects how the 2DTMS definition is interpreted, whereas the tileset CRS may determine the ordering of coordinates for data contained within the tiles), and/or the tileset CRS being a member of the CRS ensemble of the 2DTMS definition. To support dynamic CRSs, an epoch, which provides the time at which the observations were made as a fractional year, can also be specified. The standard enumerates three common data types that can be used: *map* for *map tiles* containing a visual representation of data or natural color imagery (RGB(A) pixel values), *vector* for vector features ("*vector tiles*") such as points, lines and polygons, and *coverage* for *coverage tiles* containing the actual values of measured or observed properties, often in a gridded form. The data type property is intended to be extensible for additional types of geospatial data that can be tiled, such as point clouds or 3D meshes.

In addition to these essential elements, the tileset metadata may contain other useful information. For tilesets not spanning the full extent of the area defined by the 2DTMS, limits for each tile matrix can be specified in terms of the minimum and maximum column and row for which data is available. For tiles containing vector or coverage data, a schema for the properties found within the tiles can be specified, including the name, data type and semantic information for each property. Since vector tiles in particular may contain multiple data layers (e.g., linear road features, building footprints and amenity points for street maps), this schema for the properties is provided for one or more data layers. The dimensionality of the geometry for each data layer can also be specified, where 0 is used for points, 1 for curves such as line strings, 2 for surfaces such as polygons, and 3 for solids such as polyhedrons. The minimum and maximum resolution at which a particular data layer will be found in the files can also be expressed. Additional metadata elements such as a title, a description, keywords, a short attribution, license, a point of contact, dates of creation and last update, the style used to generate the tileset and a center point location representative of the tileset are also defined and should be accurately populated to the extent possible. Whether intended for offline or online use, providing well-defined metadata for a tileset is a ***Best Practice*** that makes it significantly more interoperable, allowing clients to unambiguously know how the tiles are organized, rather than make assumption about a particular tiling scheme being used.

Brief history of 2DTMS and their use in OGC WMTS and OGC GeoPackage

2D Tile Matrix Sets were first described as such in the OGC *Web Map Tile Service (WMTS)* Implementation Standard (OGC 07-057r7), an offshoot of the OGC Web Map Service (WMS, also ISO 19128 Web map server interface) taking inspiration from the OSGeo Tile Map Service specification to allow efficient caching of map requests constrained to predefined tiles, as well as the ability to use static pregenerated tilesets. This approach followed a proven practice adopted by several commercial implementations of map services that was not yet standardized. Annex E of the WMTS Standard defines four *Well-known scale sets* (WKSS) which formed the initial basis for common 2DTMS. Three of them are based on the OGC CRS84 WGS84 geographic CRS, which is equivalent to EPSG:4326 except for having a longitude, latitude axis order. The first of these WKSS uses rounded scale denominator values, the second has scales coinciding with the round pixel sizes of common global products, whereas the third is divided as a quad-tree. The fourth WKSS is based on the spherical ("Web") Mercator projection (EPSG:3857), popular for Web mapping applications. The concept of a "fixed" 2DTMS was first used in the OGC WMTS Simple Profile. This profile defines the 2DTMS with an id of *WorldWebMercatorQuad* corresponding to the Web Mercator WKSS, with tile matrix identifier 0 being the scale with the largest denominator (a single tile for the whole world), and subsequent integers identifying the following scales. The 2DMS ids *WorldCRS84Quad* or *InspireCRS84Quad* in turn correspond to the CRS84 quad-tree WKSS, with tile matrix identifier -1 being the scale with the largest denominator, and subsequent integers used for the following scales.

OGC GeoPackage (OGC 12-128r18) is another standard which adopted the use of 2DTMS to store imagery and raster maps. The OGC GeoPackage extension for Tiled Gridded Coverage Data also allows to store 2D coverage tiles in GeoPackage.

While WMTS and GeoPackage introduced and popularized the 2DTMS concepts, an independent specification for 2DTMS did not yet exist. Version 1.0 of the 2DTMS standard was published in 2019, providing a stand-alone definition of the logical model and encodings. In addition to the XML encoding used in WMTS, the Standard also introduced a JSON encoding for defining a 2DTMS. A new requirements class for variable width tile matrices was introduced in this Standard. In addition to the well-known scale sets inherited from 2DTMS, this first version of the Standard also featured informative annexes for some commonly used 2DTMS definitions, representing a first step towards the idea of defining a common 2DTMS independently of a particular service or dataset. The two common 2DTMS from the Simple Profile were notably part of the annex, but the -1 scale was omitted from the definition of the *WorldCRS84Quad* 2DTMS, as the -180°S..180°N latitude range caused significant confusion. The *WorldWebMercatorQuad* was also renamed to simply *WebMercatorQuad* in this annex. The *GNOSISGlobalGrid* was included as an example definition of a 2DTMS using the variable width capability.

After publication of 2DTMS version 1.0, the need to unambiguously reference common 2DTMS through a URI, to register 2DTMS definitions with an authority and to maintain the definitions independently of the standard became apparent. The OGC Tile Matrix Set register was set up for that purpose. Version 2.0 of the Standard clarified the use of a URI to uniquely identify a registered 2DTMS, and introduced the TileSet metadata model and encoding, which can reference a registered 2DTMS by URI. A list of differences between the two version is detailed in the version 2.0 release notes. Version 2 should be used for all purposes, since version 1 will likely be deprecated.

In the OGC Vector Tile Pilot initiatives, additional GeoPackage extensions (VTP2 Engineering Report: OGC 19-088r2) related to tiling were prototyped. Several of these draft extensions related to the capabilities to store tiled vector data inside GeoPackage. Another extension aims to fully support the 2DTMS Standard inside GeoPackage. The 2DTMS logical model defined as part of the GeoPackage differs slightly from the WMTS model. One limitation is that it cannot support tile matrices such as the *WorldCRS84Quad* level -1 defined in the *WMTS Simple Profile.* Another limitation is the lack of support for variable width tile matrices. Both of these limitations are removed by this draft 2DTMS extension. Efforts are on-going to complete and standardize these extensions.

Tiling maps, imagery and gridded coverage data

As used in WMTS and GeoPackage, rectangular tiles of imagery or raster maps, consisting of RGB(A) pixels ready to be visualized on a display device, are well established for efficiently storing and visualizing this type of data. These tile images can be encoded in various formats, but PNG (ISO 15948) and JPEG (ISO 10918) are the most popular as they are the formats officially supported by both WMTS and GeoPackage and are also readily supported in all web browsers. However, both of these formats suffer from limitations. Neither of them supports including georeferencing information. JPEG offers a good compression ratio, but does not support translucency (which is particularly useful to tile map layers separately and allowing to combine them), and suffers from significant compression artifacts especially at higher compression ratios. PNG on the other hand does support translucency as well as lossless compression, but the compression ratio is much lower. GeoTIFF (OGC 19-008r4), itself based on TIFF (see also ISO 12639 – TIFF/IT), can also be used to store imagery and raster maps and makes it possible to georeference the data. Newer alternatives such as JPEG XL (ISO 18181), which supports translucency, a higher compression ratio for equivalent quality and lossless capabilities, are worthy of considerations despite not being as widely adopted yet.

For storing tiles of 2D gridded coverage of data values representing measured or observed properties, GeoTIFF is a common data format. It is also possible to store these values using 16-bit PNG, in particular with a scaling factor and an offset allowing to quantize real numbers to a 16-bit integer. Both of these formats are supported by the GeoPackage Gridded Coverage extension. While GeoTIFF is mostly limited to two-dimensions, netCDF can be used to encode gridded coverages in 3D, 4D or higher dimension coverages.

In addition to the possibility of encoding individual tiles as GeoTIFF, the Cloud Optimized GeoTIFF (COG) Standard (OGC 21-026) can also efficiently store large images with an internal tiling structure, allowing to efficiently access relevant spatial portions of the image. With overview images, a downsampled version of the data can be also stored and accessed when a large area but lower resolution is required. Tiles of interest from within a COG can also be easily accessed by web clients using HTTP range requests, which is a ***Best Practice*** that is particularly appealing for serving tiles directly from object storage offered by cloud-computing services, which is significantly more affordable than running a custom service requiring computation capabilities.

Tiling vector features

Tiles of data representing vector features, commonly known as *vector tiles*, present several advantages over serving a rasterized version of the same data and have seen a surge in adoption with the advent of the Mapbox Vector Tiles (MVTs) specification. Geospatial information systems have however benefited from tiling vector data since the early days of the Canada Geographic Information System (CGIS) developed in the 1960's, allowing to efficiently access and process vector data at arbitrary scales, despite hardware constraints on memory and computing resources. As well as taking up less data to encode compared to a raster equivalent, a vector tile can be queried directly for the attributes of its features and can be rendered by a visualization client in a variety of styles without needing to make any additional request to the server.

In addition to clipping vector features within a tile's geographic boundary (with the exception of an additional buffer for MVTs discussed below) and preserving only those features falling within this boundary, the features of all tiles but those of the most detailed zoom level also undergo cartographic generalization, making them suitable for display at a larger scale, as they consist of fewer points. Whereas for lines and polygons the simplification can be done purely on the basis of segment length or area, the reduction of point datasets requires filtering based on attributes to decide which points to keep at the lower zoom levels (unless dealing with dense point clouds, discussed in the next section below). A similar scale-based selection of features may also optionally be done for line and polygon features.

For the purpose of rendering vector tiles of polygon features without unwanted visual artifacts when their edges are drawn with a stroke, as well as to facilitate recombining tiled polygons into the original features, it is useful to identify the artificial segments that were introduced at the tile edges. The approach used for MVTs is to clip the geometry with a buffer that extends slightly beyond the tile's actual geographic boundary. Another approach is to flag these artificial segments in the encoding. Although MVTs are a popular encoding for vector tiles, alternative encodings exist such as FlatGeobuf or GeoJSON, and new encodings may still be devised with their own advantages.

Similar to COG, PMTiles is a generic tile container that supports accessing individual tiles within a tileset file using HTTP range requests, regardless of the nature of the content, which makes it a ***Best Practice*** for serving vector tiles from object storage.

A popular misconception regarding MVTs is that the vector data has been rasterized and thus their use is strictly limited to visualization, which is not the case. In order to efficiently encode vector data, in MVTs, the coordinates of geometry are quantized to a grid at a resolution preserving enough accuracy for the intended size, corresponding to the tile zoom level, at which the features will be rendered on the display (the accuracy is typically 1/16th of a pixel when zoomed in close on the features). The encoding instructions organized as *MoveTo* (absolute origin) and *LineTo* (relative offsets, resulting in small values achieving a good compression ratio) are also often misunderstood as "pen drawing instructions", further contributing to the misconception. In addition to being an efficient vector data exchange format, vector tiles are also an efficient storage mechanism, for example combined with the (draft) vector tiles extensions for OGC GeoPackage.

Tiling point clouds

Point clouds are datasets containing a very large number of 3D points, each with their own (x, y, z) coordinates, usually obtained through either laser scanning (LiDAR) devices and/or photogrammetry. Depending on the particular point cloud format, version and the capabilities and method of the acquiring device(s), each point often includes additional information such as the intensity, current return, number of returns, scan direction, and red, green and blue values. Some point clouds will also include a classification category (such as vegetation, road surface, ground, building...), after having gone through a classification process, which is usually the first step before applying any additional processing to the data.

Tiling also enables handling large point cloud datasets of arbitrary density by setting an upper bound on the number of points that can be contained within a single tile. For lower resolution tiles covering a larger area, the number of points can be reduced while maintaining desired properties of the data for its intended purpose by applying a clustering or filtering algorithm.

The LAS format and its compressed LAZ counterpart are efficient and popular formats for storing point cloud data which are also suitable for storing tiles of point cloud data. Similar to Cloud Optimized GeoTIFF, the Cloud Optimized Point Cloud (COPC) format provides a cloud-friendly container format based on LAZ organized as a clustered octree which can support HTTP range access. The Point Data Abstraction Library (PDAL) provides several capabilities operating on point cloud datasets, including tiling functionality.

Point clouds are particular in that they overlap considerably with both the concept of feature collections (of *Point* and/or *MultiPoint* features), as well of non-gridded coverages, while also inherently being 3D data. As such, the mechanisms, APIs and tools to process, exchange and render those data types may be applied to point cloud data as well, but the defining aspect of point clouds being the massive number of points in a dataset might prevent their use if they were not specifically optimized for dealing with point clouds.

Tiling 3D data

Other types of 3D geospatial data besides point cloud also benefit from tiling. Large scale photorealistic 3D environments can be captured, processed, curated, exchanged and visualized in tiled 3D formats. Two OGC community standards, OGC 3D Tiles (OGC 22-025r4) and OGC Indexed 3D Scene Layer (I3S) (OGC 17-014r9), define formats to exchange multi-resolution arbitrary large 3D scenes. Although both of these formats describe point clouds as a possible payload, they also support textured 3D meshes as a lighter mechanism to describe and render 3D environments. Through additional processing, laser-captured point clouds can be segmented and turned into 3D meshes, which can also be split into tiles of multiple resolutions. Both I3S and version 1.0 of 3D Tiles organize data as a Bounding Volume Hierarchy (BVH), a tree structure for which higher resolution nodes (tiles) can be accessed and displayed to replace or augment coarser nodes, as a viewer gets closer. Contrary to the *2D Tile Matrix Set* (2DTMS) approach whereas space is tiled independently of content, these BVHs are often organized around the density of the content, so as to distribute data in tiles containing approximately the same amount of data. It is however possible to organize a BVH based on a fixed tiling scheme such as a 2DTMS. OGC 3D Tiles 1.1 (also referred to as "*3D Tiles Next*") supports the concept of what it calls implicit tiling, which more closely follows the same approach as the 2D Tile Matrix Set standard, whereas space is tiled independently of content, which also corresponds to the approach used in the OGC CDB data store standard used for simulation and training environments. While a few types of tiling schemes are supported such as quad trees or octrees, 3D Tiles 1.1 does not yet support some 2D Tile Matrix Sets such as those using variable width tile matrices. While the original 3D Tiles standard defined the *Batched 3D Models (.b3dm)* tile payload as consisting of a Khronos glTF™ 3D model prefixed by a feature table header, 3D Tiles Next supports glTF directly as the tile payload. While 3D Tiles 1.1 is much more similar to the approach to tiled 3D content suggested in the 2D Tile Matrix Set standard and planned as a requirements class of the *OGC API – 3D GeoVolumes* candidate standard (draft OGC 22-029), some additional complexity remains in requiring to explicitly declare 3D transforms for each tile and declare tile availability, which requires a client to request and continuously parse potentially several different tileset metadata files and transformation for individual tiles. By contrast, the approach suggested in 2D Tile Matrix Set Annex J relies on a single tileset metadata file, just like 2D map tiles and vector tiles, where the implied transform for any given tile can be inferred simply by its matrix, row, column identifier, and tile availability is gathered based on *full* and *empty* hints while retrieving parent tiles. It is possible that a future version of 3D Tiles will better realign these two standards while simplifying the implementation of 3D Tiles clients.

Accessing data of interest efficiently through OGC API standards

The OGC API family of standards defines a modernized approach to deliver geospatial data and provide processing capabilities as REST Web APIs which can be described and documented using an API description language such as the OpenAPI Specification. The OGC API standards encompass the traditional functionality of the legacy OGC Web Services such as the Web Mapping Service (WMS), the Web Map Tile Service (WMTS), the Web Feature Service (WFS, OGC 09-025r2 also ISO 19142), the Web Coverage Service (WCS, OGC 17-089r1) and the Web Processing Service (WPS, OGC 14-065). A key aspect of the OGC API standards is that they are defined in a modular manner, consistently reusing API building blocks such as schemas, responses and query parameters. Several common resource paths are also defined by the *OGC API – Common* standard, with Part 1 (OGC 19-072) defining resources for a landing page, the API documentation and description and conformance declaration, while Part 2 (draft OGC 20-024) defines resources for listing available collections of data and retrieving the description of an individual collection, including providing links to one or more data access mechanisms. This commonality provides a cohesive framework by which a single Web API deployment may implement several alternative access mechanisms for the same dataset, for example allowing to retrieve maps through *OGC API – Maps* (draft OGC 20-058), vector features through *OGC API – Features* (OGC 17-069r4 – also ISO 19168 Geospatial API for features), as well as both vector tiles and map tiles through *OGC API – Tiles* (OGC 20-057). Similarly, data from a coverage collection may be retrieved using *OGC API – Coverages* (draft OGC 19-087), while coverage tiles are also available through *OGC API – Tiles*.

Unlike its predecessor WMTS which only defined the retrieval of map tiles, *OGC API – Tiles* allows the retrieval of raw data tiles, such as vector tiles and coverage tiles, which are more bandwidth efficient, can be rendered in multiple styles by the client, support query operations without server round-trips and also enable data analytics use cases. The *OGC API – Tiles* Standard is organized into a set of modular requirements classes, with the Core defined as simply the ability to retrieve tiles from a URL template supporting *level*, *row* and *column* variables as defined by the *2D Tile Matrix Set and Tileset metadata* standard. The *Tileset* class is defined as the ability to retrieve tileset metadata, also described in the *2DTMS&TSMD* Standard, including support for its JSON encoding. Additional classes define the ability to list multiple available tilesets, establish a connection with the *OGC API – Common* landing page and/or collection resources, allow the selection of specific collections to include as part of the tiles, allow specifying a time of interest, and identify support for specific encodings such as PNG, JPEG, GeoTIFF, netCDF, Mapbox Vector Tiles and GeoJSON.

For 3D content, the candidate *OGC API – 3D GeoVolumes* Standard (draft OGC 22-029) defines an access mechanism whereas a Bounding Volume Hierarchy may be exposed supporting either or both 3D Tiles and I3S content. *3D GeoVolumes* also defines another access mechanism whereas the content for a tile addressed using a 2D Tile Matrix Set extended into three or more dimensions can be requested directly. Examples of responses for such a request could be a glTF 3D model, or a vector tile of points instantiating shared 3D models.

Most OGC API data access mechanisms support query parameters to perform requests pertaining to a specific area, time and/or resolution of interest. For *OGC API – Tiles*, as with *OGC API – Discrete Global Grid Systems*, the area and resolution of interest is inherently tied to an identifier which is part of the resource URL and follows a predefined partitioning scheme. This facilitates efficient caching for both clients and servers.

Axis-aligned Discrete Global Grid Systems based on 2D TileMatrixSets

Discrete Global Grid Systems (DGGSs), as defined by the OGC Abstract Specification Topic 21 (OGC 20-040r3, also ISO 19170), partition the entire Earth in multi-resolution hierarchical grids. Topic 21 also defines a sub-category of DGGSs qualified as equal-area requiring the area variation between zones of a given type and hierarchy level to remain below a threshold of 1%. DGGSs have many applications including effective sampling, parallel processing, access and visualization of data. Several of the applications and concepts of DGGS overlap with those of tiles. Depending on the rigor with which the mathematical definition of a tile is applied, the zones of a DGGS grid can in fact be considered as tiles. Unlike 2D Tile Matrix Sets, where tiles are always rectangular, the geometry of DGGS zones can be of different shapes depending on the particular DGGS. For example, most of the zones of an ISEA3H (Icosahedral Snyder Equal Area aperture 3 Hexagonal) DGGS have a hexagonal shape, except for twelve zones at each hierarchy level which are pentagonal.

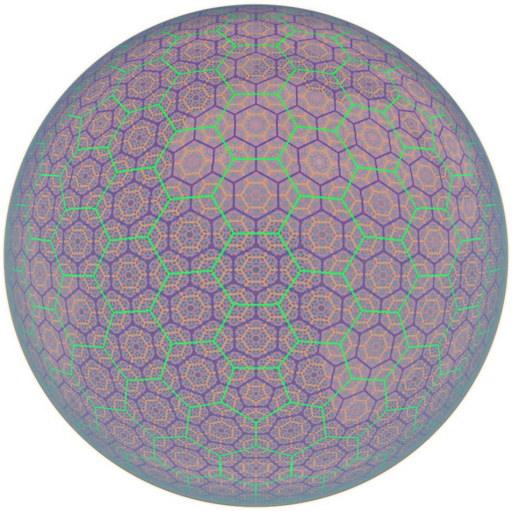


Figure 5. Level 3,4,5,6, and 7 of ISEA3H DGGS

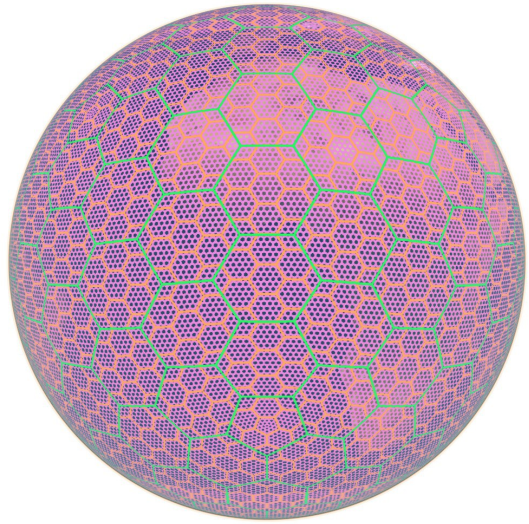


Figure 6. Level 3,5, and 8 of ISEA3H DGGS

However, most hierarchical 2D Tile Matrix Sets fully covering the Earth can form the basis for defining a DGGS. These DGGSs based on a 2D Tile Matrix Set also conform to the *Part 4: axis-aligned zones* DGGS extension to Topic 21 (draft OGC 20-049), which requires all zone geometry edges to be aligned with the axes of the coordinate reference system in which the grid is defined. These axes may be the latitude and longitude axes, such as with the *GNOSIS Global Grid* Tile Matrix Set defined in the WGS84 EPSG:4326 geographic CRS. While an equal-area axis-aligned grid could potentially be defined in this geographic CRS, the GNOSIS Global Grid does not qualify as such due to exceeding the maximum 1% area variance. However, through the variable width capability, the grid tends towards equal area, and maintains a variance between the zone area and the median below 50% up until a very detailed zoom level, including at the poles.

Two other DGGS based on a 2D Tile Matrix Set are of a particular interest. The ISEA9R (Icosahedral Snyder Equal Area aperture 9 Rhombic) DGGS is a dual of ISEA3H, defined in the same ISEA (Icosahedral Snyder Equal Area) projection, where the centroid of every ISEA3H hexagon and pentagon at even levels corresponds to the vertex of a rhombus. The ten root rhombuses at level 0 correspond to pairs of triangles of the ISEA projection's icosahedron solid. By defining the axes of the CRS as following the edges of these root rhombuses, which is equivalent to applying a 60° clockwise rotation and a 30° skew to the projection defined as the unfolded icosahedron plane, the rhombuses are transformed into rectangles, allowing to define a 2D Tile Matrix Set where level 0 consists of 5 x 6 tiles, 10 of which being valid and corresponding to each root rhombus. Because the DGGS based on this 2D Tile Matrix Set is both equal area and axis-aligned, because it maintains a direct relationship with its dual ISEA3H hexagonal DGGS which is favored for the sampling advantages of hexagonal zones, and because its zones have a simple rhombic shape (or rectangular, if considering the transformed axes) which is both simple to index and for which to encode data in formats such as GeoTIFF, ISEA9R is an appealing DGGS choice.

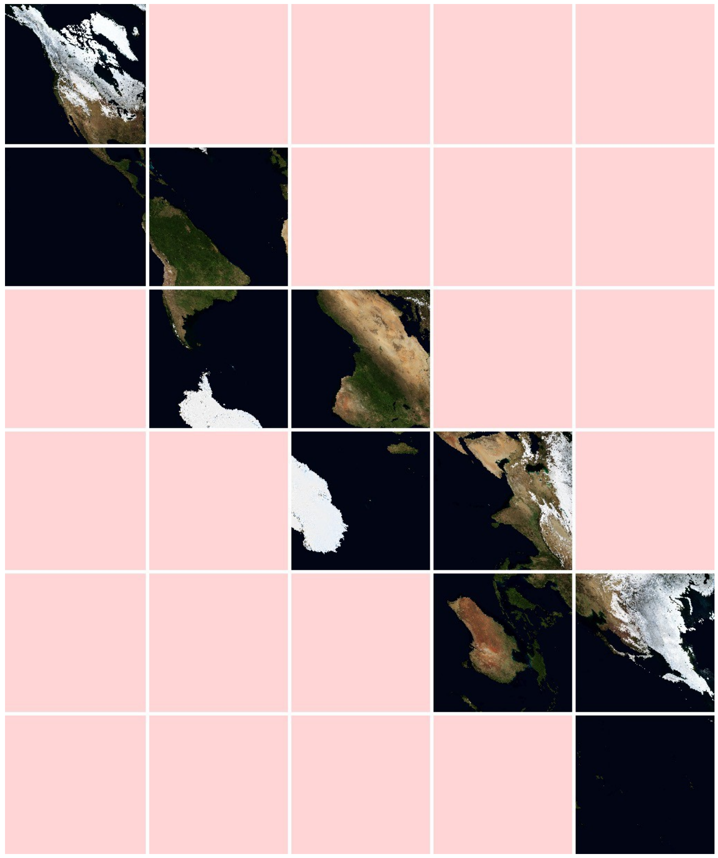


Figure 7. Level 0 of ISEA9R 2DTMS and axis-aligned DGGS,
a dual DGGS of ISEA3H even levels (only 10 out of the 30 tiles in the matrix are valid)

Another DGGS which can be described as based on a 2D Tile Matrix Set and is both equal-area and axis-aligned is the rHEALPix DGGS, based on the rHEALPix projection. The rHEALPix projection is itself based on the HEALPix projection, where the four triangles of each polar region at level 0 have been combined into a single square zone, which is the same shape as the four equatorial zones, resulting in a 2DTMS with 6 valid zones out of 12 (4 x 3) at this first level of the hierarchy.

The *OGC API – DGGS* candidate Standard (draft OGC 21-038) defines requirements classes for clients sharing an understanding of a particular DGGS with a server to either retrieve data for an area of interest, effectively asking "*What is here?*", or to perform spatial queries to answer questions of the type *"Where is it?"*. For a DGGS which can be described as a 2D Tile Matrix Set, the zone data retrieval functionality is mostly equivalent to the *OGC API – Tiles* access mechanism. However, zone data retrieval extends to other DGGS such as ISEA3H where the zone geometry is not rectangular.

The zone query requirements class on the other hand provides, for all types of DGGSs, a novel and efficient mechanism to perform spatial queries and return a list of zones, to a desired precision level, that can be compacted by replacing a full set of children zone by their parent zones, therefore only requiring small zones to detail the edges of the returned result set. This functionality is most useful in conjunction with a query language such as the OGC Common Query Language (CQL2) (draft OGC 21-065) which can express the criteria whether to return a region or not, for example by performing data integration across multiple datasets, and even multiple servers and/or processes. The compact zone list representation is also an ideal mechanism to exchange a region of interest and perform lazy evaluation while distributing parallel work across multiple nodes.

Tiles as a mechanism to facilitate parallel and on-demand processing

Partitioning geospatial data as tiles (or DGGS zones) can be used for processing purposes, either to work around memory constraints where the entire dataset cannot fit in memory, allowing to process individual tiles one at a time, or to speed up processing by distributing individual tiles across CPU or GPU cores or network nodes. While this use of tiles may not be as well-known as its visualization purposes for web client applications, this approach is key to big data analytics that a web client may trigger through an API to visualize the results. Tiles allow dealing with arbitrarily large amount of geospatial data and is common in machine learning applications as well as in high performance parallel and distributed processing. Except for the output of map rendering process, the tiled data bundles would normally contain raw data values, such as vector features, point clouds or gridded coverages.

The *OGC API – Processes – Part 1: Core (*OGC 18-062r2*)* Standard defines operations to execute a process over the web, while its *Part 3: Workflows and Chaining* (draft OGC 21-009) extension allows to chain a process as an input to another process. Part 3 also integrates data collections, as defined by *OGC API – Common – Part 2*, that may be used as inputs and/or produced as outputs of processes. These collections may be virtual, enabling on-demand processing for an area, resolution and time of interest based on OGC API data access mechanisms, such as *OGC API – Tiles* and *OGC API – DGGS*.

Accessibility considerations

The use of raw data tiles, such as vector tiles and gridded coverage tiles, made possible by *OGC API – Tiles*, enables possibilities to improve the accessibility of Web maps. Shifting the rendering of the map to the client facilitates the use of alternate rendering styles such as high contrast, using larger fonts, or alternative color schemes avoiding ambiguity for people with a particular type of color blindness. The same data expressed as coverage or vector tiles only needs to be retrieved once by the client, while also typically requiring less bandwidth to transfer compared to map tiles.

Furthermore, these data tiles are not restricted to a visual representation, as the actual values and feature information contained within them is in its raw form, which can be presented to the user in a manner suitable for them. For example, text normally rendered on a map as labels can be read aloud to the user while touching a certain area of a touch device. The same technique could also work with dynamic tactile braille devices. A similar device could potentially reproduce a height map, or heat map, in a similar way with small height variation, or even possibly temperatures, that can be sensed by touch. By providing efficient access to the raw data, the OGC APIs opens up a world of possibilities to Web clients, which could see the introduction of several new innovative ways to convey map information to improve accessibility.

It should be noted that it would likely also be possible to infer the raw values from a traditional rendered map representation through the use of artificial intelligence, though this would involve considerably more computing resources and is less reliable than receiving the source values directly. Still, it may facilitate making a wide range of content usable with these same accessibility devices by enabling interoperability with services that have not yet transitioned to supporting client-side rendering. Such a map recognition process could for example be implemented as a process made available through *OGC API – Processes – Part 1* also supporting *Part 3*, allowing to take as input a collection from an *OGC API – Maps*, *OGC API – Tiles* (returning only map tiles), or a WMS or WMTS service end-point, and return on-demand the vector tile or coverage tile equivalent for any source map tile. The resulting tile of raw values would be suitable for these devices producing accessible representations.

Glossary

Tile (2DTMS: 4.12)

a specific region of space, which may or may not result from a tessellation, or a piece of spatial data corresponding to that subset of space

Tiling Scheme (2DTMS: 4.17)

a specific way to partition space into tiles

Coordinate System (2DTMS 4.3)

a set of mathematical rules for specifying how coordinates are to be assigned to points

Coordinate Reference System (2DTMS 4.2)

a coordinate system that is related to the real world by a datum

(2D) Tile Matrix Set (2DTMS: 4.14)

a specific type of tiling scheme (tiling *space* i.e., a Coordinate Reference System) as defined in the 2DTMS standard using rectangular tiles

(2D) Tile Matrix (2DTMS 4.13)

a single matrix of a Tile Matrix Set, usually corresponding to a different scale / zoom-level

Tile Set (2DTMS 4.16)

a set of data tiles produced by tiling a dataset (or a subset of that dataset) according to a particular Tiling Scheme (whether a 2DTMS, or another type of tessellation e.g., ISEA3H hexagonal / pentagonal zones as typical of several Discrete Global Grid Systems, or a Bounding Volume Hierarchy as typical of *OGC 3D Tiles* tilesets)

Discrete Global Grid (DGGS 4.12)

set of zones (non-overlapping tiles, not necessarily rectangular) at the same refinement level, that uniquely and completely cover a globe

Discrete Global Grid System (DGGS) (DGGS 4.13)

integrated system comprising a hierarchy of discrete global grids, spatio-temporal referencing by zonal identifiers (indexing mechanism mapping identifiers to zones) and functions for quantization, zonal query, and interoperability