

Testbed-12 Vector Tiling Implementation Engineering Report

Table of Contents

1. Introduction	6
1.1. Scope	6
1.2. Document contributor contact points	6
1.3. Future Work	6
1.4. Foreword	6
2. References	7
3. Terms and definitions	8
3.1. vector tile tiled vector vectile	8
4. Conventions	9
4.1. Abbreviated terms	9
5. Overview	10
6.	11
6.1. Status Quo	11
6.2. Requirements Statement	11
7.	12
7.1. Targeted Solutions	12
7.2. Recommendations	13
8. Tile pyramid	14
8.1. Goal	14
8.2. Approach towards a vector tile pyramid	14
8.3. Example	15
9. Tile Encoding	16
9.1. Tile Encoding approach using GeoJSON	16
9.1.1. Generating and using clipped tiles	17
9.1.2. Example	17
9.2. Influences	27
9.2.1. GenaMap	27
9.2.2. LuciadFusion	27
9.2.3. TileStache	27
Appendix A: Revision History	28
Appendix B: Bibliography	29

Publication Date: 2017-05-15

Approval Date: 2017-01-26

Posted Date: 2016-12-29

Reference number of this document: OGC 16-067r4

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-A068>

Category: Public Engineering Report

Editor: Daniel Balog, Robin Houtmeyers

Title: Testbed-12 Vector Tiling Implementation Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is an OGC Public Engineering Report created as a deliverable of an initiative from the OGC Innovation Program (formerly OGC Interoperability Program). It is not an OGC standard and not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR'S sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Abstract

This OGC Testbed 12 Engineering Report (ER) discusses the topic of implementing vector tiles in an OGC GeoPackage. This report builds on the general topic of vector tiling discussed in OGC Testbed 12 Engineering Report [OGC 16-068r4].

Since its public release in 2012, OGC GeoPackage has been getting increasingly popular within the geospatial industry for a variety of use cases, such as a means to package geospatial data for use on a mobile device and as a means to exchange geospatial data between two systems.

The OGC GeoPackage standard currently specifies requirements (rules) for storing raster tiles and vector (simple) features. This Engineering Report proposes an extension to the supported data types by introducing an implementation for vector tiles.

While tiling and the use of multiple levels of details are a proven technique for accessing and visualizing raster data, it is less commonly applied for vector data. This is due to the increased complexity compared to raster tiling and lack of standardization on the topic. Yet, implementing vector tiles can provide the same benefits as for using raster tiles.

- Services can easily cache tiles and return them instantly upon request, without the need for any additional pre/post processing. Consequently, clients can get tiles very fast, ensuring fast and responsive maps.
- Using tiled, multileveled data representations, clients can always access the data most suitable for their current map location and scale. This avoids the need to load too much data, which can cause excessive memory usage and reduce overall performance.

The goal is to enable systems to use OGC GeoPackage as a means to store and access vector tiles in an efficient way, similar to raster tiles.

Business Value

The OGC GeoPackage standard has been beneficial for the industry by defining a standardized, interoperable approach to easily package and exchange raster tiles and vector features for use on mobile devices and other platforms.

As vector tiles provide the same benefits as raster tiles in terms of data access, there is an increasing interest by the geospatial community to define a standard way for vector tiles. From an interoperability perspective, a logical next step is to build on top of the presented work and have a discussion and agreement within OGC on the general approach towards vector tiling. This could in turn lead to the definition of an extension to the OGC GeoPackage standard. Standardizing this at the OGC level could give a boost to the adoption of vector tiles, leading to interoperable solutions capable of accessing large vector datasets.

What does this ER mean for the Working Group and the OGC?

In 2012, the OGC successfully released the GeoPackage standard to store and access vector features and raster data. The standard has been embraced by the geospatial industry and community and its popularity is increasing. A logical next step is to extend this work to support of vector tiling. By defining a standardized approach to vector tiling, the OGC can play an important role in increasing the adoption of vector tiling and improving the interoperability between solutions developed by the geospatial industry and community.

How does this ER relate to the work of the Working Group?

This ER relates to the GeoPackage SWG because the ER describes an approach of a discussion of implementing vector tiles in GeoPackage. Additionally, it relates to any working group dealing with vector feature modeling; as such, any follow-up vector tiling discussions should preferably happen in an OGC-wide context - possibly in a dedicated working group.

Keywords

ogcdocs, testbed-12, vector, tiling, GeoPackage

Chapter 1. Introduction

1.1. Scope

This OGC® Testbed 12 Engineering Report discusses a proposed implementation of vector tiles for the OGC GeoPackage standard. This ER relates to the Testbed 12 Vector Tiling Engineering Report [OGC 16-068r4], which provides a general introduction to the topic of vector tiling, independent from an implementation.

No Change Requests have been identified or defined in the context of this document.

1.2. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Robin Houtmeyers	Luciad
Daniel Balog	Luciad
Frederic Houbie	Luciad
Peter De Maeyer	Luciad
Rob Cass	Compusult

1.3. Future Work

The OGC Testbed participants involved in the vector tiling activity expect that the findings documented in this Engineering Report and the related OGC Vector Tiling Engineering Report [OGC 16-068r4] will be incorporated in future vector tiling standardization work related to OGC GeoPackage.

1.4. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- OGC: OGC 06-121r9, OGC® Web Services Common Standard (2010)

NOTE: This OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

- OGC: OGC 12-128r11, OGC GeoPackage Encoding Standard 1.01 - With Corrigendum (2015)
- OGC: OGC 12-128r12, OGC GeoPackage Encoding Standard 1.1 (2015)
- Mapbox: Mapbox Vector Tile Specification, <https://github.com/mapbox/vector-tile-spec>
- Migurski, M., et al: TileStache, <http://tilestache.org/doc/#vector-provider>
- Various: Ramer-Douglas-Peucker algorithm, https://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm
- IETF: The GeoJSON Format, https://datatracker.ietf.org/doc/rfc7946/?include_text=1 (2016)

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply. In addition, the following terms and definitions apply.

3.1. vector tile | tiled vector | vectile

packet of geographic data, packaged into a pre-defined roughly-square shaped "tile" for transfer over the web.

Chapter 4. Conventions

4.1. Abbreviated terms

- COTS Commercial Off The Shelf
- WFS Web Feature Service
- WMS Web Map Service
- WMTS Web Map Tiling Service
- JSON JavaScript Object Notation
- GeoJSON Geospatial JSON

Chapter 5. Overview

This OGC Testbed 12 Engineering Report discusses an implementation of vector tiles in OGC GeoPackage. Chapter 6 of this Engineering Report (ER) begins with a discussion of the current state and the requirements related to a vector tiling implementation. Chapter 7 discusses possible solutions and gives a high-level overview of the solution implemented in Testbed 12. This is followed by an overview of how the implementation integrates vector tiles into the tile pyramid concept defined by OGC GeoPackage (Chapter 9) and a description of the actual vector tile encoding used in the implementation (Chapter 10).

Chapter 6.

6.1. Status Quo

The technique of tiling applied to raster(ized) data has existed for over a decade and has been widely adopted and implemented by the geospatial community. This includes work in the standards community. Examples within the OGC include the Web Map Tiling Service (WMTS) and GeoPackage standards.

Tiling applied to vector data has existed over a similar amount of time, but has been less commonly applied when compared to raster tiling. However, the topic did receive increasing attention over the past several years. Most known to the general public are Google Maps and Apple Maps, which both use vector tiles to have high-quality graphics at arbitrary zoom levels. Additionally and perhaps familiar to the developer community, MapBox released a public specification for vector tiles a few years ago, with several implementations available in both COTS and open-source components. These examples all have a focus on the visualization use case in common.

OGC GeoPackage Encoding Standard version 1.0.1 [2] defines requirements and rules for storing image raster tiles and vector features. An extension for elevation raster tiles [3] is in development.

6.2. Requirements Statement

The primary focus for this Engineering Report is describing an implementation of vector tiles in an OGC GeoPackage. Special attention is given to aligning the work with the existing raster tiling approach and raster tiling schemes used by OGC GeoPackage.

Chapter 7.

When dealing with a general requirement of being able to handle large vector datasets in OGC GeoPackage, for Testbed 12 two solutions were identified:

- Introduction of a vector tile pyramid concept, similar to the existing raster tile pyramids; and
- Extension of the existing feature capabilities with the ability to represent multiple resolutions of a feature's geometry.

The first solution was chosen for work in Testbed 12. The reason is to enable the capability to handle vector data with geometries of arbitrary detail, and although having multiple resolutions addresses the use case of accessing such data at lower levels of detail, tiling is needed to enable scalable access to large geometries at high levels of detail. The second solution has also received attention outside Testbed 12, within the OGC GeoPackage SWG and also within the Defence Geospatial Information Working Group (DGIWG) [4].

A vector tile pyramid solution can be implemented in many ways. Following the general vector tiling overview from OGC Engineering Report 16-068 [1], we can identify two high-level approaches to addressing the use of vector tiles in a GeoPackage:

- Render-based tiling, which is primarily targeted towards visualization; and
- Feature-based tiling, which is targeted both towards visualization and feature access.

This Engineering Report describes a feature-based tiling solution to implement vector tiles in an OGC GeoPackage. The following section provides a high-level overview of this solution and the rationale for that solution. A render-based tiling approach was not considered in Testbed 12. This was due to the desire to have access to the original feature's geometry. Further, a solution for the visualization use case exists in the industry: the MapBox Vector Tile Specification [5]. The MapBox solution is not based upon the OGC GeoPackage approach. However, as it essentially defines a tile encoding format, it could rely on OGC GeoPackage as data container as a data source.

7.1. Targeted Solutions

This document describes a feature-based tiling solution for implementing vector tiles in an OGC GeoPackage, based on the existing standards foundation for raster tiles (see [5], Section 2.2) and using GeoJSON as tile encoding format.

The choice for a feature-based tiling is largely based on one of the primary target use cases for using a GeoPackage: supporting mobile device users in disconnected or limited network connectivity environments.

For such use cases, we see added value in giving the user direct access to the feature's geometry, without the need to go through an additional service. One example use case from OGC Testbed 12 is a client-side off-road routing calculation. The calculation is based on data residing in an OGC GeoPackage. Having the feature's geometry makes it possible to use the features within route calculations, for instance to avoid swam areas and dense vegetation. Then the same data could be used for other calculations without generating a new GeoPackage.

Having the feature's native geometry available also makes the vector tile projection-independent; a client can project the coordinates according to the desired geographic projection. It should be noted that the GeoJSON standard generally does not recommend the use of a CRS other than the default (WGS 84) ([6], section 4); however, alternative reference systems can be used if all involved parties have a prior agreement.

Finally, the reuse of the existing OGC GeoPackage concepts for raster tiles and the use of the widespread GeoJSON format for the tile encoding favors easy adoption of the proposed implementation approach for vector tiles.

7.2. Recommendations

The existing support for vector data in OGC GeoPackage does not address the use case of handling large geospatial vector datasets in applications. The Testbed 12 Vector Tiling activity attempted to define an initial implementation to support this use case. According to a DGIWG questionnaire circulated in 2015 and presented during the OGC Technical Committee meeting in Boulder, June 2015, vector tiling was one of three high-priority extensions requested for the OGC GeoPackage [4]. Therefore it is recommended to further revising the suggested implementation approach and work to evolve the specification into an official OGC GeoPackage extension. A primary recommendation is to favor the featured-based tiling approach, providing users with feature access capabilities - even in disconnected or limited network connectivity requirements.

Additionally, this ER recommends further investigating alternative solutions for storing multiple resolutions of features in OGC GeoPackage. Both can be considered as complementary solutions, each with their targeted use cases (see also Engineering Report OGC 16-068 [1]).

Chapter 8. Tile pyramid

This section discusses the implementation aspects of storing, accessing and describing vector tile pyramids in OGC GeoPackage.

The OGC GeoPackage standard identifies several tables and concepts related to raster tiles: tile matrix, tile matrix set, zoom levels, and so on. The Vector Tiling participants investigated how these existing concepts align with the requirements for vector tiles.

The primary focus of this section is on the tile pyramid modeling. Details about the tile encoding itself are described in the next [Tile Encoding](#) section.

This and the following section assume knowledge of OGC GeoPackage concepts such as tile matrix and tile matrix set. Readers that are unfamiliar with these concepts are directed to the latest [GeoPackage Encoding Standard](#).

8.1. Goal

The goal of the implementation experiment was to encode a dataset as vector tiles and then store the encoded tiles in a GeoPackage. The approach was to follow as close as possible the process for storing raster tile sets.

The rationale for storing the vector tile sets in this manner was to capitalize on the widely adopted practice for storing and accessing raster data in a tiled way and to support clients that already have extensive code bases for this in place. In essence, retrieving a set of vector tiles for an area at a particular level of detail should be no different than retrieving a set of raster tiles.

For the experiment, the target data set was a collection of California trail data provided by the USGS in the Esri Shapefile format. This trail data was a subset of the data accessible from [USGS National Transportation Dataset \(NTD\) for California 20160401 State or Territory Shapefile](#).

8.2. Approach towards a vector tile pyramid

This section discusses the approach followed to create a vector tile pyramid from the USGS dataset.

A GeoPackage tile matrix set (tile pyramid) is defined consisting of the following.

- A set of bounds in a destination Coordinate Reference System (CRS). In this case, the min/max bounds would be "-124.233665948855" "32.5577547848806" "-115.615938218481" "48.9998552322758" for the CRS as defined by urn:ogc:def:crs:OGC:1.3:CRS84.
- A tile size in pixels, 256px x 256px.
- A set of tile matrices from level 0 to n. The resolution of these tile matrices was determined by using the pixel size for 256x256 tiles for the tile set. The starting pixel size at level 0 is 1.40625 in the x and y dimension.

Vector tiles are stored in a tile pyramid whose descriptive tables are the same as the tables for raster tiles, with the following notes.

- In the `gpkg_contents` table, the `data_type` value for the tile matrix table is set to "vectortiles".
- For relevant entries in the `gpkg_tile_matrix` table, `tile_width`, `tile_height` are still set to pixels with an implicit CRS-based coordinate. This coordinate is derived from interpreting the tile matrix where the clipping coordinates for any vector tile are derived from interpolating the pixel offsets for any tile.
- In the content table containing the tiles, tiles are encoded as GeoJSON in the `tile_data` column. The precise nature of this encoding is described in the next [Tile Encoding](#) section of this ER.

The purpose of having multiple levels of detail for vector tiles is to provide generalized / simplified views of the data at a small scale, where features are simplified in complexity or completely hidden if smaller than a pixel at that level of detail. The extreme case is the tile at `z=0, x=0, y=0` where there is only one feature visible after thinning and generalization is performed. When processing the original dataset, a starting and ending level of detail was decided upon. This was based on visibility of any features after generalization and the practicality of continuing beyond a certain scale where all features should be visible. There is no need to encode vector tiles at the level where all data would be generalized away and there is no need to unnecessarily encode vector tiles beyond a level of detail where the number of nodes is manageable by clients.

As these constraints will vary from dataset to dataset, it is up to the producer to determine the start and stop for the levels of detail. For the test data, the zoom levels chosen were from 0 to `n`.

The production process begins creating tiles at the largest scale, starting at the upper left corner for the tile matrix. For each tile, the data set is queried for features that intersect with the tile. Each feature is clipped to the bounds of the tile, followed by a generalization step if required.

NOTE

For the sample dataset, the Douglas-Peucker-Ramer generalization algorithm described [here](#) was used. This was the algorithm of choice for the experiment. Mileages may vary depending on the algorithm chosen and the dataset being reduced.

The reduced features are encoded as GeoJSON as described in [Tile Encoding](#) and stored in a tile pyramid GeoPackage table called `california_trails`.

8.3. Example

The following table shows an example row from the `california_trails` table:

Table 2. california_trails

id	zoom_level	tile_column	tile_row	tile_data
30262	17	21069	51612	<pre> {"features":[{"id":"Trans_TrailSegment.387","properties":{"AnchorTile":"21069,51613,17"},"type":"Feature","geometry":{"crs":{"properties":{"name":"EPSG:4326"},"type":"name"},"type":"LineString","coordinates":[[-122.12986516,38.24066162],[-122.129853,38.241253],[-122.130625,38.242254],[-122.131155,38.243256],[-122.13140998,38.2434082]]}}],"type":"FeatureCollection"} </pre>

Chapter 9. Tile Encoding

This chapter discusses the approach followed to encode a tile in the vector tile pyramid described in the previous section. The first subsection discusses the implementation approach. This approach relies on GeoJSON as encoding format. The next subsection discusses influences that helped refining this approach.

9.1. Tile Encoding approach using GeoJSON

For Testbed 12, a vector tile encoding was developed as an extension of GeoJSON. This method is developed to satisfy a set of general requirements defined up-front:

- Well known encoding for easy adoption;
- No repetition in data to minimize tile sizes;
- Usable by web clients;
- Capable of representing vector features, and not solely vector drawing instructions; and
- Works with coordinates relative to the defined Coordinate Reference System (CRS) instead of coordinates relative to the tile.

GeoJSON satisfies the requirements that the information is usable in web clients, it is a well known standard and a GeoJSON encoding compresses well.

The resulting vector tile format assumes that the tiles are generated by clipping a common feature collection. The following list defines the format in more detail.

- The vector tiles are composed of standard GeoJSON FeatureCollections, as defined in the [GeoJSON format specification](#).
- The vector tiles are expected to conform to a tile matrix set based upon the tile organization used in the OGC WMTS and GeoPackage standards.
- Any multi-node feature geometry is composed of the geometry type, the CRS and the clipped and simplified coordinates.
- Non-geometric feature properties (aka, attributes) are found only in a feature's anchor tile. This anchor tile represents the tile at the lowest scale containing the feature's first geometry point. The expectation is that if clients require this sort of feature information, the client will inspect the JSON representation of the feature, look for the anchor tile reference [x,y,z] in the properties and load that tile from the common tile source based on these coordinates. An example of an anchor tile can be found in the remainder of this document.
- All other tiles for a particular feature contain a property that specifies the feature anchor tile for reference/lookup.
- Anchor tiles are [x,y,z] referenced by means of the feature property AnchorTile.
- Precision of double values in the geometries is kept to 6 decimal places. More precision is questionable for data produced for the source scale of 1:24000. Producers would need to judge the best precision for their data set.

9.1.1. Generating and using clipped tiles

As described in the previous section, the vector tile format assumes that the tiles are generated by clipping a feature collection. The nature of clipped geometries requires that certain strategies are used for re-constructing complete geometries from the clipped components. One challenge that arises with clipping is that some original geometry vertices are co-incident with the edge of the tile and are not distinguishable from clipped vertices. Another challenge is that artificial vertices are introduced as a clipping by-product.

The requirement for accurate vertices would only need to be satisfied when performing analysis on the geometry and when there is a requirement to use more vertices of the feature beyond one tile.

In the case where an vertex is artificial, it will always lie on a line and not interfere with the visual topology. However, there may be rendering cases related to the nodes along the line that require only the origin nodes to be rendered. This requires the identification of "real" vs. "artificial" vertices.

In addition to the other identified requirements, polygons require clipped vertices to capture the interior clipped points - for instance, when a polygon encompasses a tile. Also, when polygons with holes are clipped such that the hole is cut up along multiple tiles, the hole essentially disappears as it is no longer contained by a polygon. The task for the producer is to correctly close the hole with a linear ring that goes in the opposite direction to standard polygons. It is up to the client to determine a union of holes across multiple tiles.

Artificial vertices are identified by the 0-indexed position that they occupy in the GeoJSON for the line or polygon linear ring to which they belong. For complex polygons, artificial vertices are contained in a structure parallel to the organization of the geometry itself.

9.1.2. Example

The following two sections include examples of actual tile encodings. The first one is an anchor tile, containing both the geometric and non-geometric properties of the tile's features; the second one is a regular tile, containing a feature with geometric properties and a link to the anchor tile.

Anchor tile

The following vector tile illustrates an example of an anchor tile. This tile contains all the feature attributes for the feature `Trans_TrailSegment.377`. All clipped "sub-features" of this feature point to this tile as their anchor tile.

Example:

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          [
            -116.46682692,
            32.58958471
          ]
        ]
      }
    }
  ]
}
```

```

    ],
    [
      -116.55313151,
      32.66093635
    ],
    [
      -116.44092328,
      32.75266902
    ],
    [
      -116.41269946,
      32.86769808
    ],
    [
      -116.53877365,
      33.01045291
    ],
    [
      -116.46283349,
      33.09344091
    ],
    [
      -116.65601436,
      33.2824875
    ],
    [
      -116.56199245,
      33.63097298
    ],
    [
      -116.66133321047546,
      33.75
    ]
  ],
  "crs": {
    "properties": {
      "name": "EPSG:4326"
    },
    "type": "name"
  },
  "type": "LineString"
},
"id": "Trans_TrailSegment.718",
"properties": {
  "FCODE": "20600",
  "FTYPE": "206",
  "GNIS_ID": "233358",
  "LENGTH": "",
  "LOAD_DATE": "",
  "NAME": "Pacific Crest National Scenic Trail",
  "PERMANENT_": "{8C7E5053-2B8C-4FFD-B076-1B280AB952F7}",

```

```

        "SHAPE_LEN": "41.3610087166864",
        "SOURCE_D00": "Pacific Crest National Scenic Trail",
        "SOURCE_DAT": "{68FF80D8-27E1-496A-906B-3506F7E6B204}",
        "SOURCE_FEA": "",
        "SOURCE_ORI": "U.S. Forest Service",
        "TRAILNUMBE": "",
        "clipidx": "[[8]]"
    },
    "type": "Feature"
},
{
    "geometry": {
        "coordinates": [
            [
                -116.54234332,
                32.83711628
            ],
            [
                -116.64006738,
                32.76147883
            ]
        ],
        "crs": {
            "properties": {
                "name": "EPSG:4326"
            },
            "type": "name"
        },
        "type": "LineString"
    },
    "id": "Trans_TrailSegment.879",
    "properties": {
        "FCODE": "20600",
        "FTYPE": "206",
        "GNIS_ID": "",
        "LENGTH": "",
        "LOAD_DATE": "",
        "NAME": "SECRET CANYON",
        "PERMANENT_": "d61721b7-2fe2-4a8f-a00f-d5ebc0bb6db9",
        "SHAPE_LEN": "0.203716902196746",
        "SOURCE_D00": "Cleveland National Forest",
        "SOURCE_DAT": "{E794EEBA-2D7A-41E4-8C6F-80A0E7C68E8D}",
        "SOURCE_FEA": "",
        "SOURCE_ORI": "U.S. Forest Service",
        "TRAILNUMBE": ""
    },
    "type": "Feature"
},
{
    "geometry": {
        "coordinates": [

```

```

        [
            -117.5519941,
            33.71385121
        ],
        [
            -117.63662768,
            33.70267181
        ]
    ],
    "crs": {
        "properties": {
            "name": "EPSG:4326"
        },
        "type": "name"
    },
    "type": "LineString"
},
"id": "Trans_TrailSegment.1022",
"properties": {
    "FCODE": "20600",
    "FTYPE": "206",
    "GNIS_ID": "",
    "LENGTH": "",
    "LOAD_DATE": "",
    "NAME": "JOPLIN",
    "PERMANENT_": "dd00a69f-ae7d-44da-a856-b7d532f2c290",
    "SHAPE LENG": "0.175482519113874",
    "SOURCE_D00": "Cleveland National Forest",
    "SOURCE_DAT": "{E794EEBA-2D7A-41E4-8C6F-80A0E7C68E8D}",
    "SOURCE_FEA": "",
    "SOURCE_ORI": "U.S. Forest Service",
    "TRAILNUMBE": ""
},
"type": "Feature"
},
{
    "geometry": {
        "coordinates": [
            [
                -116.41829277,
                32.84168102
            ],
            [
                -116.48106693,
                32.93175826
            ]
        ],
        "crs": {
            "properties": {
                "name": "EPSG:4326"
            },

```

```

        "type": "name"
      },
      "type": "LineString"
    },
    "id": "Trans_TrailSegment.1057",
    "properties": {
      "FCODE": "20600",
      "FTYPE": "206",
      "GNIS_ID": "",
      "LENGTH": "",
      "LOAD_DATE": "",
      "NAME": "PACIFIC CREST TRAIL - THING VA",
      "PERMANENT_": "d3a2f662-7e7d-4e5f-9367-e08bf86bef2b",
      "SHAPE LENG": "0.196650080513957",
      "SOURCE_D00": "Cleveland National Forest",
      "SOURCE_DAT": "{E794EEBA-2D7A-41E4-8C6F-80A0E7C68E8D}",
      "SOURCE_FEA": "",
      "SOURCE_ORI": "U.S. Forest Service",
      "TRAILNUMBE": ""
    },
    "type": "Feature"
  },
  {
    "geometry": {
      "coordinates": [
        [
          -117.011602,
          32.849548
        ],
        [
          -116.985758,
          32.924297
        ]
      ],
      "crs": {
        "properties": {
          "name": "EPSG:4326"
        },
        "type": "name"
      },
      "type": "LineString"
    },
    "id": "Trans_TrailSegment.1229",
    "properties": {
      "FCODE": "20600",
      "FTYPE": "206",
      "GNIS_ID": "",
      "LENGTH": "",
      "LOAD_DATE": "",
      "NAME": "Sycamore Canyon",
      "PERMANENT_": "d5580f5c-d239-41f1-af94-0d9908bdc014",

```

```

        "SHAPE_LEN": "0.142819524732349",
        "SOURCE_D00": "IMBA_CA",
        "SOURCE_DAT": "{80C95A1B-429B-419E-B5D4-549B02140C22}",
        "SOURCE_FEA": "",
        "SOURCE_ORI": "IMBA/Adventure Projects LLC",
        "TRAILNUMBE": ""
    },
    "type": "Feature"
},
{
    "geometry": {
        "coordinates": [
            [
                -116.48319115,
                32.73201715
            ],
            [
                -116.41829277,
                32.84168102
            ]
        ],
        "crs": {
            "properties": {
                "name": "EPSG:4326"
            },
            "type": "name"
        },
        "type": "LineString"
    },
    "id": "Trans_TrailSegment.1242",
    "properties": {
        "FCODE": "20600",
        "FTYPE": "206",
        "GNIS_ID": "",
        "LENGTH": "",
        "LOAD_DATE": "",
        "NAME": "PACIFIC CREST TRAIL - BOULDER",
        "PERMANENT_": "90b93a62-ae22-4210-a3db-8c0b5aff6a7e",
        "SHAPE_LEN": "0.202593595438561",
        "SOURCE_D00": "Cleveland National Forest",
        "SOURCE_DAT": "{E794EEBA-2D7A-41E4-8C6F-80A0E7C68E8D}",
        "SOURCE_FEA": "",
        "SOURCE_ORI": "U.S. Forest Service",
        "TRAILNUMBE": ""
    },
    "type": "Feature"
},
{
    "geometry": {
        "coordinates": [

```



```

        -117.154093,
        32.937839
    ],
    [
        -117.222751,
        32.904826
    ],
    [
        -117.154174,
        32.937861
    ]
],
"crs": {
    "properties": {
        "name": "EPSG:4326"
    },
    "type": "name"
},
"type": "LineString"
},
"id": "Trans_TrailSegment.1420",
"properties": {
    "FCODE": "20600",
    "FTYPE": "206",
    "GNIS_ID": "",
    "LENGTH": "",
    "LOAD_DATE": "",
    "NAME": "Los Penasquitos Canyon Preserve",
    "PERMANENT_": "9097f656-a97e-4013-91dd-7fd93a4999be",
    "SHAPE LENG": "0.206173131981949",
    "SOURCE_D00": "IMBA_CA",
    "SOURCE_DAT": "{80C95A1B-429B-419E-B5D4-549B02140C22}",
    "SOURCE_FEA": "",
    "SOURCE_ORI": "IMBA/Adventure Projects LLC",
    "TRAILNUMBE": ""
},
"type": "Feature"
},
{
    "geometry": {
        "coordinates": [
            [
                -116.85497545,
                33.39620463
            ],
            [
                -116.93500937,
                33.39035528
            ]
        ]
    },
    "crs": {

```

```

        "properties": {
            "name": "EPSG:4326"
        },
        "type": "name"
    },
    "type": "LineString"
},
"id": "Trans_TrailSegment.1436",
"properties": {
    "FCODE": "20600",
    "FTYPE": "206",
    "GNIS_ID": "",
    "LENGTH": "",
    "LOAD_DATE": "",
    "NAME": "CUTCA",
    "PERMANENT_": "bd12b868-93df-4d0f-937a-bbbda59cf676",
    "SHAPE LENG": "0.108604683825053",
    "SOURCE_D00": "Cleveland National Forest",
    "SOURCE_DAT": "{E794EEBA-2D7A-41E4-8C6F-80A0E7C68E8D}",
    "SOURCE_FEA": "",
    "SOURCE_ORI": "U.S. Forest Service",
    "TRAILNUMBE": ""
},
"type": "Feature"
}
],
"type": "FeatureCollection"
}

```

Regular tile

Example:

```

{
  "features": [
    {
      "geometry": {
        "coordinates": [
          [
            -121.7169650104664,
            45
          ],
          [
            -121.70815421,
            45.03863078
          ],
          [
            -121.77517299,
            45.12566287
          ],
        ],
      },
    }
  ],
}

```

[
-121.67932797,
45.29557617
],
[
-121.86869233,
45.50948923
],
[
-121.78934898,
45.58207884
],
[
-121.83049934,
45.67492808
],
[
-121.97703143,
45.6661756
],
[
-122.04683388,
45.74370056
],
[
-121.78690337,
45.84659288
],
[
-121.76378182,
46.12161056
],
[
-121.56867219,
46.17995667
],
[
-121.50035761,
46.38420385
],
[
-121.39606124,
46.41512384
],
[
-121.46528497,
46.5119745
],
[
-121.37186983,
46.62585875

```
],  
[  
  -121.38382389,  
  46.70821995  
],  
[  
  -121.51838825,  
  46.87584412  
],  
[  
  -121.37612255,  
  47.05764836  
],  
[  
  -121.4050813,  
  47.12025836  
],  
[  
  -121.29865998,  
  47.14937623  
],  
[  
  -121.33880407,  
  47.27510744  
],  
[  
  -121.4558599,  
  47.35640778  
],  
[  
  -121.36777994,  
  47.46233494  
],  
[  
  -121.14403742,  
  47.51377513  
],  
[  
  -121.15186621,  
  47.65036712  
],  
[  
  -121.05920293,  
  47.71486467  
],  
[  
  -121.12774316378827,  
  47.8125  
]  
],  
"crs": {
```

```

        "properties": {
            "name": "EPSG:4326"
        },
        "type": "name"
    },
    "type": "LineString"
},
"id": "Trans_TrailSegment.718",
"properties": {
    "AnchorTile": "22,20,6",
    "clipidx": "[[0,27]]"
},
"type": "Feature"
}
],
"type": "FeatureCollection"
}

```

9.2. Influences

A review of existing approaches was performed during the Testbed to define a reasonable encoding; consequently, credit is due to the following approaches and sources.

9.2.1. GenaMap

In 1985, GenaMap developed a tiling approach for storing raster and vector data. The vector data was stored in a topology edge/node format. This data was separately indexed and encoded relative to the tile centre. One feature of the encoding process was to explicitly mention the edge nodes for any feature. A multi-node feature clipped to a tile bounds would contain a list of all "Artificial" edge nodes generated by the clipping. The tiling approach developed for Testbed 12 used the idea of storing the edge nodes as indices of the multi-node feature component. Storing these nodes can aid clients to know when to eliminate nodes for processing or to capture cases such as a polygon completely containing a tile.

9.2.2. LuciadFusion

In discussion with Luciad, the approach of storing an anchor tile that contains the feature attributes was introduced. The encoding for the feature attributes was designed using a Key-Value pair approach. The choice for where the anchor tile was determined to be the highest-resolution tile containing the first point of the feature.

9.2.3. TileStache

The TileStache approach was one of the first implementations of vector tiles using GeoJSON.

Reference details for the sources are given in [References](#).

Appendix A: Revision History

Table 3. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
April 14, 2016	.1	R. Houtmeyers	all	IER
August 30, 2016	.5	R. Cass	all	added description of JSON-based vector tile encoding
September 23, 2016	.7	R. Houtmeyers	all	general revision
October 20, 2016	.9	R. Houtmeyers	all	addressed comments from internal review
October 25, 2016	.9	R. Houtmeyers	all	addressed comments from external reviews

Appendix B: Bibliography

- [1] Balog, D., Houtmeyers, R., Houbie, F., De Maeyer, P.: OGC Testbed 12 Vector Tiling Engineering Report, 16-068. OGC (2016).
- [2] Daisy, P.: OGC GeoPackage Encoding Standard 1.0.1 - With Corrigendum. 12-128r11. OGC (2015)
- [3] Web: OGC: OGC GeoPackage Elevation Extension, <https://github.com/opengeospatial/geopackage-elevation/>
- [4] Web: OGC DGIWG: Proposed Extension for Multi-resolution Vector Data in OGC GeoPackage, Presentation from May 12, 2016, presented during the DGIWG Technical Panel Meetings, https://portal.opengeospatial.org/files/?artifact_id=68934
- [5] Web: MapBox: Mapbox Vector Tile Specification, <https://github.com/mapbox/vector-tile-spec>
- [6] Web: The GeoJSON Format: https://datatracker.ietf.org/doc/rfc7946/?include_text=1