Open Geospatial Consortium

Publication Date: 2015-11-19

Approval Date: 2015-07-13

Posted Date: 2015-05-28

Reference number of this document: OGC 15-067

Reference URL for this document: http://www.opengis.net/doc/PER/T11-geopackage-terrain

Category: Public Engineering Report

Editor(s): Gobe Hobona;Roger Brackin

Testbed-11 Multi-dimensional GeoPackage Supporting Terrain and Routes Engineering Report

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is 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.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

License Agreement

Contents

Page

Figures

Page

Tables

Page

Figure 1. Sequence diagram of the routing use case 9
Figure 2. A screenshot demonstrating routing (source: Compusult) 17
Figure 3. An example of a grid coverage 18
Figure 4. Example of a TIN 19

Abstract

Routing is one of the most widely used functions of mobile applications. Routing often requires consideration of a variety of factors in order to provide reasonable estimations of journey time and the cost of travel. Another widely used function of mobile applications is the visualization of characteristics of terrain such as slope or viewsheds. The goal of this engineering report is to describe the work carried out in the OGC Testbed-11 for multidimensional terrain and routing support on SQLite databases that conform to the OGC GeoPackage standard. This OGC® Engineering Report (ER) describes an approach for the storage of routing and multidimensional terrain data in such databases. The ER also presents the results and lessons learnt from the experimentation conducted by the testbed.

Business Value

This OGC Engineering Report describes approaches that could improve interoperability by enhancing the ability of different geospatial software products to exchange routing and terrain data. The content of the Engineering Report is important to achieving interoperability in location-based technologies because mobile applications have constrained processing and storage capabilities that require efficient use of storage resources.

Keywords

ogcdocs, ogc documents, testbed-11, geopackage, mobile, terrain, routing, topology, sqlite

Testbed-11 Multi-dimensional GeoPackage Supporting Terrain and Routes Engineering Report

1 Introduction

The Urban Climate Resilience (UCR) thread of OGC Testbed-11 sought to respond to the need to make climate information and related data readily available for the public and government decision makers to prepare for changes in the Earth's climate. The Routing aspect of the UCR thread set out to determine how databases based on the GeoPackage standard [1] could support routing applications. The multidimensional terrain aspect of the UCR Thread set out to explore how multidimensional terrain data could be stored in geopackages.

1.1 Scope

The goal of this OGC Engineering Report is to describe the work carried out in the OGC Testbed-11 for multidimensional terrain and routing support on SQLite databases that conform to the OGC GeoPackage standard.

In addition to routing, this Engineering Report includes:

- discussion of the role of topology in routing
- proposal for an approach for storing topological data for the purposes of supporting a routing engine
- description of an implementation of a routing engine used in the testbed

In relation to multidimensional terrain data, this Engineering Report includes:

- description of terrain data models
- review of coverage support in relational databases
- description of an approach for representing multidimensional terrain data in geopackages

This OGC Engineering Report is applicable to initiatives involving mobile computers such as handheld and wearable devices.

1.2 Background

Previous OGC testbeds explored and successfully demonstrated the creation, retrieval and update of vector and tiled raster data in geopackages. As vendors and users adopt the GeoPackage standard, the focus is now shifting to exploring standardization of how geopackages could be applied in a typical OGC-enabled mobile application. One of the most widely used functions of mobile applications is routing, which often requires consideration of a variety of factors in order to provide reasonable estimations of journey time and cost of travel. Some of those factors include characteristics of the terrain such as slope or, in the case of flooding, characteristics such as accessibility. The aim of this Engineering Report is to describe the work carried out in the testbed in relation to multidimensional terrain and routing support on geopackages, as well as the results and lessons learnt.

1.3 Document contributor contact points

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

Name	Organization
Gobe Hobona PhD.	Envitia Ltd.
Roger Brackin MSc.	Envitia Ltd.
Norman Barker	IBM
Raj Singh PhD.	IBM
Barbara Klis	Envitia Ltd.
Adel Bolbol PhD.	Envitia Ltd.
Chris Clark	Compusult
Ingo Simonis	Open Geospatial Consortium
Dave Wesloh	NGA

1.4 Future work

No improvements to this document are planned.

1.5 Forward

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.

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.

1. OGC 12-128r11, *OGC Geopackage Encoding Standard - With Corrigendum, 2015*

2. OGC 06-121r3, *OpenGIS® Web Services Common Standard*

3. OGC 12-119r1, *OWS-9: OGC Mobile Apps: Definition, Requirements, and Information Architecture, 2013*

4. OGC 14-006r1, *OGC® Testbed 10 Recommendations for Exchange of Terrain Data, 2014*

5. ISO/IEC 13249-3:2006, *Information technology -- Database languages -- SQL multimedia and application packages -- Part 3: Spatial*

6. Dijkstra, E. W., A note on two problems in connexion with graphs, Numerische Mathematik, 1, pp. 269-271, 1959

7. Geisberger R., Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks (PDF) (Thesis). Institut für Theoretische Informatik Universität Karlsruhe, 2008

8. ISO 19125-1:2004, *Geographic information -- Simple feature access -- Part 1: Common architecture*

9. ISO 19125-2:2004, *Geographic information -- Simple feature access -- Part 2: SQL Option*

3 Terms and definitions

For the purposes of this Engineering Report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [2] and in the OpenGIS® Abstract Specification shall apply. In addition, the following terms and definitions apply.

3.1
geopackage (all lowercase)
an instance of a database that conforms to the GeoPackage standard.

3.2
GeoPackage
the OGC standard for a platform-independent SQLite database file that may contain vector geospatial features, tile matrix sets of raster maps at various scales and metadata.

3.3
handheld mobile computing device
A small, portable computing device, typically having a display screen with touch input and/or a miniature keyboard and weighing less than 2 pounds (0.91 kg). A handheld computing device has an operating system (OS), and can run various types of application software, known as apps (adapted from OGC 12-119r1[3]).

3.4
interoperability
Capability to communicate, execute programs or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.

3.5
mobile application
Mobile applications (mobile apps) are software products developed for mobile devices which may include handheld devices, such as smart phones, or wearable devices, such as smart watches. These applications can be pre-installed on the devices during manufacture, downloaded by customers from various mobile software distribution platforms or delivered as web applications using server-side or client-side processing.

3.6
resource
A configured set of information which is uniquely identifiable to a user. This can be realized as in-line or external content or by one or more configured web services.

3.7
wearable mobile computing device
A mobile computer that is worn on the body and forms part of one's clothing.

4 Conventions

4.1 Abbreviated terms

BLOB	Binary large object
UCR	Urban Climate Resilience
DGIWG	Defense Geospatial Information Working Group

DTED	Digital Terrain Elevation Data
ER	Engineering Report
ESM	Elevation Surface Model
GML	Geography Markup Language
NSG	National System for Geospatial Intelligence
OGC	Open Geospatial Consortium
OpenLS	OpenGIS Location Services
SDI	Spatial Data Infrastructure
SWG	Standards Working Group
TIFF	Tagged Image File Format
TIN	Triangular Irregular Network
URL	Uniform Resource Locator
WFS	Web Feature Service
WMS	Web Map Service
WCS	Web Coverage Service
WPS	Web Processing Service

4.2 GeoPackage versus geopackage

This Engineering Report uses the term "GeoPackage" to refer to the standard and the term "geopackage" to refer to an instance of a database that conforms to the standard. This convention is intended to make it easier for readers of this report and does not reflect an official OGC convention.

5 Previous Work

OGC Testbed (Testbed 10) experimented with using a Web Processing Service (WPS) to create a geopackage and identified recommendations for the exchange of terrain data. The Testbed report made the following recommendations regarding GeoPackage support for terrain data and its relationship to the Terrain Data Exchange Model (TIXM) which has been developed by Eurocontrol for the Aviation industry:

"The use of GeoPackage for terrain data requires a new extension. Adding such an extension is supported by the format so this should not pose any problems. The same metadata as used in TIXM can be embedded in the GeoPackage file. However, GeoPackage offers a number of advantages compared to TIXM with an external data file:

- *GeoPackage is already a widely accepted OGC standard.*
- *GeoPackage is a scalable format due to its support for tiling, multi-leveling and sparseness.*
- *GeoPackage is a single self-describing file which eases distribution.*" [4].

The Testbed-10 report also made the following observations and recommendations:

"To make GeoPackage compatible with elevation data, two main components need to be extended. First, the currently [supported] raster tile formats (PNG and JPEG) are limited to Red-Green-Blue (RGB) pixels. Elevation data requires integer or floating point values for each pixel. To accommodate this, GeoPackage could be extended to support a file format that does support floating point images. An example of this would be the TIFF format. Secondly, there needs to be an extension so that you can make the distinction between color-based tiles and elevation-based tiles. Currently the assumption is that all tiles are color-based. An extension could be added to GeoPackage to allow certain tiles or tile-pyramids to be "tagged" as containing elevation data instead of color data.

Elevation data would also require a way of defining the vertical reference. For this we recommend the usage of compound EPSG-codes inside the GeoPackage file to declare both the horizontal and vertical reference. This would require no change to the GeoPackage standard." [4].

6 Methodology

To achieve the objectives for the GeoPackage activity, the experimentation conducted in Testbed-11 followed the following process:

1. Review routing and topology data sources
2. Identify data and services to support the use case
3. Configure individual components to support routing and multidimensional terrain
4. Assess interoperability and identify lessons to learn

7 Routing

There are two main approaches to routing that are typically applied in mobile computing. The first approach is the edge-node (alias arc-node) model which relies on possible routes being described in the form of a topological network. The edge-node model relies on having a valid directed graph that includes all participating nodes and edges. Network data validation involves testing for network connectivity, overlap, duplicate lines and line intersection. The second approach is known as cross country movement (CCM) and relies primarily on applying a 'drainage' model to raster terrain data.

The basic problem addressed by routing through a topological network is that of a user wanting to travel from a location A (that is at or near a node or edge on the network) to a location B (that is at or near a node or edge on the network). This basic problem can be made more complex by considering the cost of traveling along the various edges to be different; for example, travelling up a steep hill is likely to cost more (in terms of energy) than travelling downhill. Such complex routing can be used to represent a number of cost models; for example, the time taken to travel between towns or the risk of an ambush along a path of motorway.

The basic problem addressed by CCM is that of a user wanting to travel from location A to location B, between which there is no road infrastructure. In its basic form, CCM can consider the drainage model by factoring in slope and elevation between the start and end points. However, the problem can be made more complex by factoring in texture of the terrain, the type of vehicle and the locations of no-go areas.

Although the aforementioned routing approaches address the same problem under different contexts, the provision of either approach within a service-based infrastructure would be the same. That is, both the topology-based and CCM-based algorithms offer geo-computational processes that can be provided through web services conforming to the Web Processing Service (WPS) standard or the Open Location Services Interface Standard (OpenLS)[1]. The testbed therefore focused its implementation on the topology-based routing; however the findings of this report are equally applicable to CCM-based routing.

7.1 Use Case

Testbed-11 implemented a routing use case to examine the potential for using OGC web services and other standards such as GeoPackage. The testbed used Geopackages to store evacuation route alternatives. The alternatives are calculated based on the latest flooding data. For this reason, the route calculation service, which is front-ended by a WPS interface, can be parameterized with location masks indicating flooded areas that should be avoided. The location masks are taken into account during route calculation. The use case can be described as follows:

---

[1] http://www.opengeospatial.org/standards/ols

Actors:

- Mobile client
- GeoPackaging WPS
- Routing Service
- WFS of Flooded Areas
- Flood Prediction WPS
- Feature Enrichment Service
- WFS-T

Basic steps (sequence of their interactions is presented Figure 1):

1. The mobile client sends a request to the GeoPackaging WPS for a geopackage containing routes from Location A to Location B.
2. WPS sends a request to the Routing service.
3. Routing service sends a request to a WFS of flooded areas.
4. The WFS, of flooded areas, requests an update from a Flood Prediction WPS.
5. The Flood Prediction WPS obtains additional information from a Flood Enrichment Service and WFS-T.
6. The Flood Prediction WPS obtains other information from a WFS-T.
7. The Flood Prediction WPS returns the predicted flooded areas to the Flooded-area WFS for storage.
8. The Flooded-area WFS returns the flooded areas to the Routing service.
9. Routing service provides three alternative routes from Location A to Location B.
10. GeoPackaging WPS generates a geopackage from the routes and returns the geopackage to the mobile client.

The components involved in this use case and sequence of their interactions is presented Figure 1.

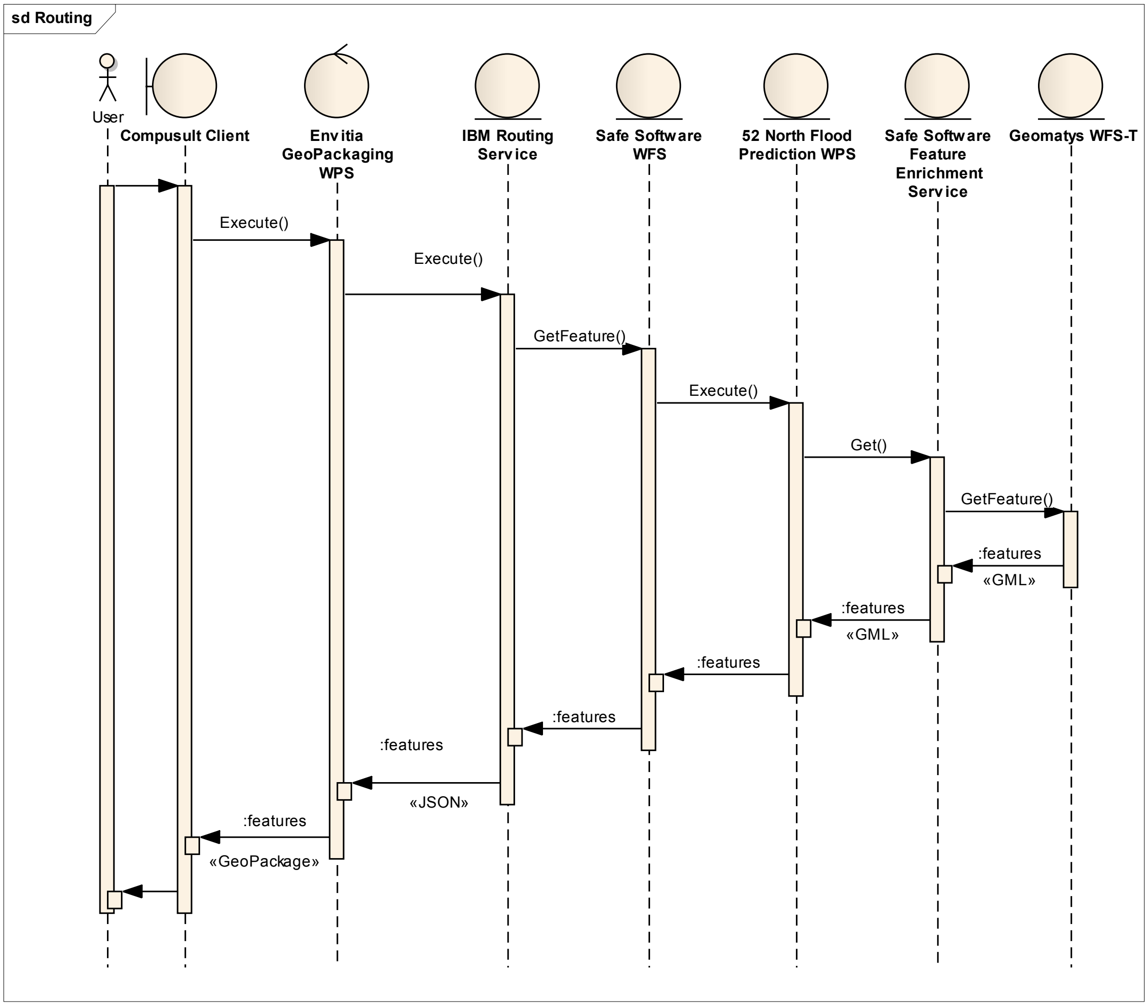

Figure 1. Sequence diagram of the routing use case

7.2 The Role of Topology in Routing

A key enabler of routing is topology. Topology[2] is the science and mathematics of relationships used to validate the geometry of vector entities, and for operations such as network tracing and tests of polygon adjacency. The fundamental topological data model deals with spatial relationships by representing spatial objects such as point, line and polygon features as a basic graph of topological elements such as nodes, edges and faces. The topological data model requires all line ends that are within a specified proximity of one another to be snapped together such that they have common coordinates. Nodes are

[2] From mathematics, topology is the study of geometric properties and spatial relations unaffected by the continuous change of shape or size of figures.

placed at those line ends, as well as where lines cross. The topological data model can be used to govern coincident geometry by enforcing how features share geometry, to support advanced editing by imposing restrictions on the data model, to execute advanced topological queries such as those for navigation and to determine spatial data integrity. Of most importance to this testbed is the support to navigation offered by the topological data model.

7.3 Relational Representation of Topological Networks for Routing

In order to meet the objectives of Testbed-11, it was necessary to explore how a routing network could be represented in a GeoPackage. The testbed participants reviewed various approaches for encoding topology as adopted by related standards organizations and open initiatives. This section presents descriptions of relevant aspects of the schema used by Spatialite (Table 1), ISO SQL/MM [5] (Table 3, Table 4 and Table 5), PostGIS (Table 6, Table 7 and Table 8) and the Open Source Routing Machine (OSRM) (Table 9 and Table 10).

Table 1. Fields from a Spatialite Roads table

column name	data type
name	text
node_from	Integer
node_to	Integer
cost	Double
length	Double
class	text
oneway_fromto	integer
oneway_tofrom	integer
Geometry	Geometry

Table 2. Fields from a Spatialite Nodes table

column name	data type
nodeid	Integer

osmid	Integer
cardinality	Integer
geometry	geometry

Table 3. ISO SQL/MM ST_Node table

column name	data type
topology	character
node_id	integer
geometry	st_point
containing_face	Integer

Table 4. ISO SQL/MM ST_Edge table

column name	data type
topology	character
edge_id	Integer
start_node	Integer
end_node	Integer
next_left_edge	Integer
next_right_edge	Integer
left_face	Integer
right_face	Integer
geometry	st_curve

Table 5. ISO SQL/MM ST_Face table

column name	data type
topology	character
face_id	integer
mbr	st_polygon

Table 6. PostGIS Edge table

column name	data type
edge_id	character
start_node	integer
end_node	integer
next_left_edge	integer
next_right_edge	integer
left_face	integer
right_face	integer
Geom	geometry

Table 7. PostGIS Node table

column name	data type
node_id	integer
containing_face	integer
geom	geometry

Table 8. PostGIS Face table

column name	data type
face_id	integer
mbr	geometry

Table 9. OSRM Node table

column name	data type
latitude	integer[3]
longitude	integer[4]
node_id	integer
flags	integer

Table 10. OSRM Edge table

column name	data type
source_node	integer
target_node	integer
edge_length	integer[5]
direction	integer

[3] multiple by 1,000,000 and round to integer

[4] multiple by 1,000,000 and round to integer

[5] unit: m; MUST be > 0

edge_weight	integer
edge_type	integer
street_index	integer
flags	integer

7.4 Proposed Tables for Supporting Routing in GeoPackage Files

In order to support routing in geopackages the testbed participants defined new tables to describe edges and nodes. An illustration of the tables is shown below and the SQL CREATE statements are presented in the Appendix.

Table 11. Proposed gpkg_topology_nodes table for GeoPackage

column name	data type
id	integer
geometry_table	integer
geometry_id	integer
containing_face	integer

Table 12. Proposed gpkg_topology_faces table for GeoPackage

column name	data type
id	integer
geometry_table	integer
geometry_id	integer

Table 13. Proposed gpkg_topology_edges table for GeoPackage

column name	data type
id	integer
start_node	integer
end_node	integer
start_end_flow	integer
end_start_flow	integer
classification	text
length	double
cost	double
next_left	integer
next_right	integer
left_face	integer
right_face	integer
geometry_table	integer
geometry_id	integer

7.5 Implementation

Topological networks represented according to an edge-node model can be applied in routing calculations using shortest path algorithms such as Dijkstra's Algorithm or Contraction hierarchies. OSRM is an example implementation of both these algorithms and their use of an edge-node topology model.

Dijkstra's algorithm is designed to find the length of an optimal path between two vertices in a graph [6]. The algorithm maintains a set of nodes and their tentative distances from each other. Each node is visited in the order of its distance to the source node. When a node is visited, the distance of the tentative target node is set to the length of the path from source node via the visited node through to the tentative target node - provided that this leads to a cost improvement. The cost can be calculated from characteristics of the edge; for example, the distance or slope. Alternatively, the cost can be calculated from factors external to the edge; for example, weather, day of the week or some other factor.

Contraction hierarchies are an approach for replacing segments from the input graph with shortcuts to a next level in a hierarchy of networks [7]. This involves pre-processing prior to the actual routing. In some cases, such as with OSRM, Contraction hierarchies are used alongside Djikstra’s algorithm.

To understand how the edge-node relational representation of topology supports routing, the testbed installed a modified version of the OSRM. Supported by data from OpenStreetMap, OSRM is a C++ implementation of a routing engine for shortest paths in road networks. OSRM offers operations such as location of nearest node to a given coordinate, location of nearest point on any street segment for a given coordinate and computation of shortest path between two coordinates given an ordered list of via points.

The OSRM normalized format, which is used for describing road networks ingested into an OSRM engine, structures information about edges and nodes according to a schema that closely resembles the relational models presented in Section 7.3. This observation highlights the potential for standardizing representation of the edge-node model in GeoPackages.

The testbed deployed the OSRM engine and modified it to offer a service endpoint based on the WPS. In its most basic form, a WPS-enabled routing service requires a start and end point to invoke a route calculation to answer the question, "find me a route from Location A to Location B". More complexity can be inserted by including one or more waypoints into a request to answer the question, "find me a route from Location A to Location B via C, D and E".

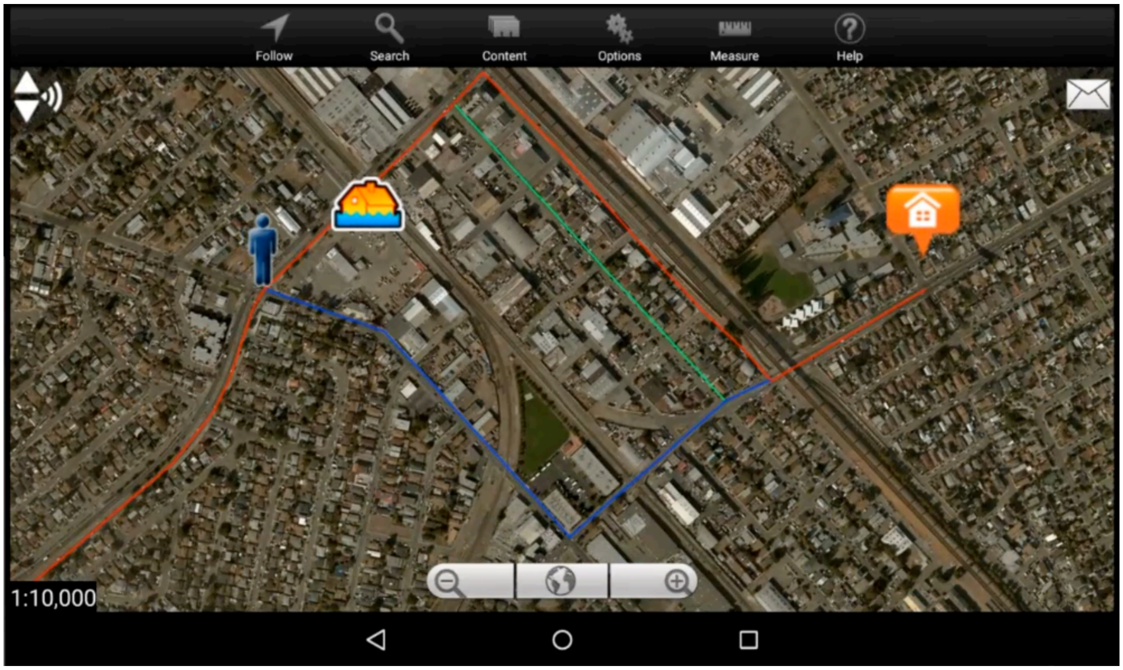

Figure 2. A screenshot demonstrating routing (source: Compusult)

The testbed implemented an approach to achieving the recalculation of routes based on external factors such as obstructions (e.g. floods level) by enabling the routing engine to accept masking parameters as input. A screenshot from the testbed demonstration is shown in Figure 2. The figure shows the primary route in red, the location of a house known to be flooded and an alternative route (in blue) to avoid the flooded location. The flood-avoiding route has been calculated based on the location and size of masking points (i.e. the location known to be flooded). The screenshot also shows an alternative route that does not consider the flooded location (green).

8 Multidimensional Terrain

8.1 Introduction

Terrain data is a fundamental requirement for a variety of geospatial analyses. Applications of terrain data include calculations of drainage in civil engineering, viewshed analysis in military planning, line of sight estimation in law enforcement and several other key aspects of geospatial technologies. Historically, these operations were mostly run from desktop systems, with readily available processing and storage capacity. The ubiquity of mobile devices has led to demand and expectation from users that some, if not all, of the aforementioned terrain analyses will be available on handheld mobile devices.

In Testbed-10 participants discussed the exchange of terrain data in GML, GeoPackage and other formats. OGC Testbed 11 builds on the achievements of the previous testbed by expanding the examination of terrain support in geopackages to examine potential support for multidimensional terrain data.

8.2 Terrain Data Models

Terrain data in geospatial applications may be held as coverages in formats such as GeoTIFF, GMLJP2 or similar formats. OGC standards acknowledge a number of different structures for holding coverages, namely coverage grids, triangular irregular networks (TIN) and point coverages.

Coverage grids are the most common geometric representation of data used to create elevation models. A grid is a tesselation of space into areas, referred to as cells, defined by a consistent pattern. The pattern most commonly used in terrain data products is that of a rectangular grid consisting of cells defined by four adjacent points. An illustration of a grid coverage is presented in Figure 3.

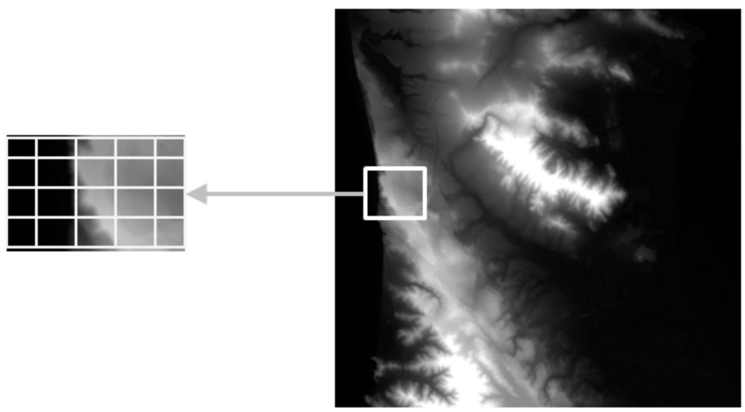

Figure 3. An example of a grid coverage

As the name implies, a TIN is an irregular collection of points that form a set of non-overlapping triangles that share edges to create a continuous surface. Figure 4 presents an example of a TIN composed of triangles with vertices formed from four-dimensional (4D) coordinates, namely latitude and longitude shown by the position of each vertex, elevation shown through the black-formatted (decimal) labels and a confidence measure shown through blue-formatted (integer) labels.

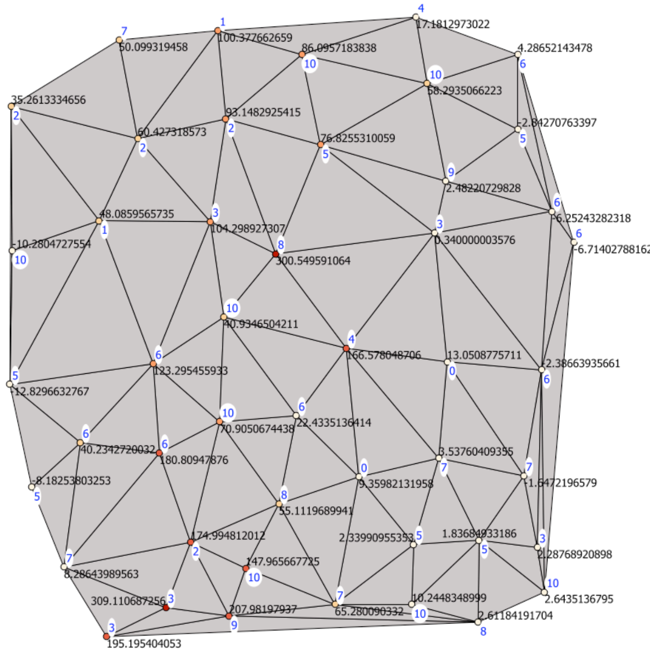

Figure 4. Example of a TIN

A point coverage represents a set of elevation values assigned to a set of arbitrary points, with each point identified by a pair of horizontal (X,Y) coordinates and assigned one or more elevation values as attributes. The nodes of the triangulation presented in Figure 4 are an example of a point coverage.

In some cases coverages modelled according to one approach (e.g. grids) can be interpolated from coverages modelled according to another approach (e.g TIN). This engineering report will examine GeoPackage support for each type of coverage separately.

8.3 Review of Coverage Support in Relational Databases

The current GeoPackage standard offers support for tiled raster data that is primarily intended for presentation (i.e. maps rendered on tiled images). This section describes current GeoPackage support for tiled raster data. The section also describes approaches used by a selection of other relational databases for packaging terrain data. The approaches examined included Rasterlite[6] and PostGIS Raster[7]. The former is the raster

[6] http://www.gaia-gis.it/gaia-sins/rasterlite-docs/rasterlite-man.pdf

[7] http://postgis.net/docs/manual-2.1/RT_reference.html

data capability offered by Spatialite and the latter is the raster data capability offered by PostGIS.

8.3.1 GeoPackage Raster

The GeoPackage standard offers an approach for storing raster tiles to form a multi-resolution pyramid structure. Four specific tables play the key role in storage of such data, namely gpkg_tile_matrix_set, gpkg_tile_matrix, gpkg_contents and a user-named table that contains the actual binary encoded tiles. This section describes each of these tables.

The gpkg_tile_matrix_set table defines the minimum bounding rectangular area (i.e. box) and references the spatial reference system used for all content in a tile pyramid user data table.

Table 14. GeoPackage gpkg_tile_matrix_set table

column name	data type
table_name	text
srs_id	integer
min_x	double
min_y	double
max_x	double
max_y	double

The gpkg_tile_matrix table documents the structure of the tile matrix at each zoom level in each tile's table. Both rectangular and square tiles are supported by this approach. The approach adopted by this table allows zoom levels of different resolution to be represented.

Table 15. GeoPackage gpkg_tile_matrix table

column name	data type
table_name	text
zoom_level	integer
matrix_width	integer

matrix_height	integer
tile_width	integer
tile_height	integer
pixel_x_size	double
pixel_y_size	double

The standard explains the purpose of the gpkg_contents table as being to provide identification and descriptive information for an application to display to a user. The gpkg_contents table is used for both vector and raster data.

Table 16. GeoPackage gpkg_contents table

column name	data type
table_name	text
data_type	text
identifier	text
description	text
last_change	datetime
min_x	double
min_y	double
max_x	double
max_y	double
srs_id	integer

A uniquely named user data table stores each tile matrix set, with the tile data stored as Binary Large Objects (BLOBs).

Table 17. GeoPackage user-named table

column name	data type
id	integer
zoom_level	integer
tile_column	integer
tile_row	integer
tile_data	blob

8.3.2 Rasterlite

Rasterlite is a library designed to offer read and write access to raster datasets in Sqlite databases. The library is part of the Spatialite suite of tools for working with geospatial data in Sqlite databases.

The raster data are stored in the Sqlite database as BLOBs. A table with the name of the raster stores the raster data in a column of type BLOB named raster. A representation of the table is shown in Table 18.

Table 18. Rasterlite 'rasters' table

column name	data type
id	integer
raster	BLOB

When raster data is stored in an Sqlite database using Rasterlite, the rasters are stored as tiles in a pyramid structure. A metadata table, with a structure illustrated in Table 19, stores the identifiers and characteristics of each tile.

Table 19. Rasterlite 'metadata' table

column name	data type
id	integer
source_name	text

tile_id	integer
width	integer
height	integer
pixel_x_size	double
pixel_y_size	double
geometry	geometry

8.3.3 PostGIS Raster

PostGIS Raster stores datasets using a minimal data structure that consists of a single type and is stored in a single table. A representation of the table is shown in Table 20. The database allows a single table row with a column of type raster to correspond to one tile. As illustrated in Table 20, the raster data are stored in the database as BLOBs.

Although offering a minimal table structure for storing a raster dataset, PostGIS Raster offers several functions for retrieving metadata about the raster dataset. This approach differs from that used by Rasterlite and GeoPackage which store the metadata in separate columns. PostGIS raster uses the underlying GDAL[8] library to process the raster and thus extract the metadata after the raster has been stored. For raster Sqlite databases based on mobile devices, such an approach would not be appropriate because of the load that processing an image would impose on a mobile device.

Table 20. PostGIS Raster table

column name	data type
rid	integer
rast	raster[9]
filename	text

[8] http://www.gdal.org/

[9] BLOB-encoded data type

8.4 Approaches for Supporting Multidimensional Terrain Data in Geopackage

To explore how multidimensional terrain data could be represented in a GeoPackage, OGC Testbed 11 generated three types of multidimensional terrain datasets and explored ways through which such terrain datasets could be stored in a GeoPackage. The following sub sections describe each proposed approach.

8.4.1 Grid Coverage

The testbed found the tile raster support offered by the current version of the GeoPackage standard to be capable of supporting a multidimensional terrain dataset, subject to a few minor revisions to the standard.

The first issue is that the standard currently specifies only PNG and JPEG as the default formats supported. Other formats can be supported through the extension mechanism supported by the standard. Therefore it is possible to extend a GeoPackage to support GeoTIFF while maintaining conformance to the standard.

Another issue was support for grid collections which could be used to represent a multidimensional grid by grouping together multiple single-tile coverages. To represent a multidimensional grid in a GeoPackage, the testbed explored whether a collection of grid coverages could be associated through a collection dataset.

The following tables describe mappings to GeoPackage tables from WCS Coverage descriptions. All of the XPaths are relative to wcs:CoverageDescription.

Table 21. Mappings to gpkg_tile_matrix_set from WCS Coverage Descriptions

geopackage field	coverage description
table_name	…/wcs:CoverageId
srs_id	…/gml:domainSet/gml:RectifiedGrid/gml:origin/gml:Point@srsName
min_x	…/gml:boundedBy/gml:Envelope/gml:lowerCorner(0)
min_y	…/gml:boundedBy/gml:Envelope/gml:lowerCorner(1)
max_x	…/gml:boundedBy/gml:Envelope/gml:upperCorner(0)
max_y	…/gml:boundedBy/gml:Envelope/gml:upperCorner(1)

Table 22. Mappings to gpkg_tile_matrix from WCS Coverage Descriptions

geopackage field	coverage description
table_name	…/wcs:CoverageId
zoom_level	"0"
matrix_width	"1"
matrix_height	"1"
tile_width	…/gml:domainSet/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high(0)
tile_height	…/gml:domainSet/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high(1)
pixel_x_size	…/gml:domainSet/gml:RectifiedGrid/gml:offsetVector
pixel_y_size	…/gml:domainSet/gml:RectifiedGrid/gml:offsetVector

Table 23. Mappings to gpkg_contents from WCS Coverage Descriptions

geopackage field	coverage description
table_name	…/wcs:CoverageId
data_type	"tiles"
identifier	…/wcs:CoverageId
description	[BLANK]
last_change	[BLANK]
min_x	…/gml:boundedBy/gml:Envelope/gml:lowerCorner(0)
min_y	…/gml:boundedBy/gml:Envelope/gml:lowerCorner(1)
max_x	…/gml:boundedBy/gml:Envelope/gml:upperCorner(0)
max_y	…/gml:boundedBy/gml:Envelope/gml:upperCorner(1)
srs_id	…/gml:domainSet/gml:RectifiedGrid/gml:origin/gml:Point@srsName

Table 24. Mappings to the user-named table from WCS Coverage Descriptions

geopackage field	coverage description
id	“1”
zoom_level	“0”
tile_column	“0”
tile_row	“0”
tile_data	[GeoTIFF binary]

8.4.2 TIN

By definition, a TIN is an irregular collection of points that form a set of non-overlapping triangles that share edges to create a continuous surface. To test the potential to support TINs, the testbed generated TINs using the Delaunay Triangulation function in QGIS. A GeoPackage was then generated from the resulting TIN dataset using multidimensional Well Known Text (WKT) geometries. When inserted into a geopackage, a geometry described in WKT in an SQL statement is converted into a Well Known Binary (WKB) and stored as a BLOB. A sample of the multidimensional geometries used in Figure 4 is presented below.

```
POLYGON ((-122.448557856 37.610676648 207.98197937000 9, -122.385857760
37.609119759 2.61184191704 8, -122.479419762 37.606659334
195.19540405300 3, -122.448557856 37.610676648 207.98197937000 9))
```

WKT and WKB are specified in Part 1 [8] and Part 2 [9] of the ISO 19125 standard. Both are limited to four dimensions, labelled as X, Y, Z and M[10]. The testbed therefore concluded that representation of a TIN could already be supported through the vector data support offered by the GeoPackage standard. However, the dependency of GeoPackage on WKT and WKB means that the GeoPackage geometries support a maximum of four dimensions.

[10] This engineering report refers to the dimensions collectively as XYZM.

8.4.3 Point Coverage

A point coverage represents a set of elevation values assigned to a set of arbitrary points, with each point identified by a pair of horizontal (X,Y) coordinates and assigned one or more elevation values as attributes. To test potential support for this type of coverage, the testbed converted point coverage data represented using XYZM coordinates into a vector GeoPackage. The testbed found that this type of coverage could already be supported through the vector data support offered by the GeoPackage standard. A sample of the points used is presented below:

```
POINT (-122.450623448 37.649558651 70.9050674438 10.0000000000)
POINT (-122.450568470 37.727925980 100.377662659 1.00000000000)
POINT (-122.400561027 37.730384607 17.1812973022 4.00000000000)
POINT (-122.385857760 37.609119759 2.61184191704 8.00000000000)
POINT (-122.370838349 37.624007521 2.28768920898 3.00000000000)
POINT (-122.385461331 37.625539699 1.83684933186 5.00000000000)
POINT (-122.498030903 37.636673867 -8.1825380325 5.00000000000)
```

9 Conclusions

This Engineering Report describes the work carried out by the UCR thread in OGC Testbed 11 to investigate the potential support for routing and multidimensional terrain data in SQLite databases that conform to the OGC GeoPackage standard. The report has also presented the results and lessons identified from the experimentation conducted in Testbed 11. Based on the findings of the UCR thread, this Engineering Report concludes that the current version of the GeoPackage standard allows for the storage of routing and multidimensional terrain data. The Engineering Report further concludes that there is a need to standardize how geopackages support both of these types of data - through use of the existing extension mechanism of the GeoPackage standard rather than through a major change to the standard.

9.1 Recommendations

The following recommendations are based on the work done in Testbed-11:

9.1.1 Development of a topology extension for geopackages to support routing

Testbed 11 demonstrated how a routing engine that ingests data modelled according to an edge-node model can provide routing information within an OGC service infrastructure. The testbed also identified opportunities for extending the GeoPackage standard to support topological networks that such routing engines rely one. This engineering report therefore recommends that the OGC develops an extension to the GeoPackage standard, using the approach described in Section 7.4, to standardize how topological networks should be represented in a GeoPackage using the edge-node model.

9.1.2 Development of a multidimensional terrain data extension for geopackages

This Engineering Report describes how multidimensional terrain could be stored in a GeoPackage by storing multidimensional coverages as single-tile coverages. The Testbed activity produced an example of such a GeoPackage to demonstrate the concept. The recommendation isthat the OGC develops an extension to the GeoPackage standard, using the approach described in Section 8.4.1, to standardize how multidimensional terrain data should be stored in geopackages.

10 Annex A

Date	Release	Editor	Primary clauses modified	Description
8/5/2015		Carl Reed	Various	Preparation for publication