

## Principles and Applications of a Multiresolution Geodata Cube

Lassi Lehto, Jaakko Kähkönen and Juha Oksanen

Geoinformatics and Cartography  
Finnish Geospatial Research Institute (FGI)  
National Land Survey of Finland  
Masala, Finland

email: [lassi.lehto@nls.fi](mailto:lassi.lehto@nls.fi), [jaakko.kahkonen@nls.fi](mailto:jaakko.kahkonen@nls.fi), [juha.oksanen@nls.fi](mailto:juha.oksanen@nls.fi)

**Abstract**—The Data Cube concept provides a useful metaphor for management of raster geodata resources in the cloud. An initiative, called GeoCubes Finland, has been launched with the aim to facilitate access to geospatial raster data for academic research. The work is carried out in the context of a major research infrastructure development program in Finland. In the ingestion process, data sets are pre-processed into a harmonized, multiresolution cloud-based repository and brought into a common georeferencing frame, resolution levels, encoding format and tiling scheme. A custom Application Programming Interface (API) has been developed for flexible query, download and analysis of the repository's content layers. Cloud-optimized access to pre-stored resolution levels supports efficient interactive visual exploration of analysis results computed on-the-fly. The presented application examples of the GeoCubes repository include field inspection, route finding and wind exposure analysis.

**Keywords**—*raster data; multi-resolution; harmonisation; cloud service; visualization.*

### I. INTRODUCTION

Use of raster-formatted data sets in geospatial analysis is increasing rapidly. At the same time geographic data are being introduced into disciplines outside the traditional domain of geoinformatics, like climate change, intelligent transport, health care and immigration studies. Geospatial data sets are also growing rapidly in number and volume. In particular, the volume of raster data sets is becoming difficult to manage. The resolution of image sensors has steadily improved and new imaging technologies have been taken into use. In addition, integrating geospatial raster data sets for analysis has become a tedious task, as the data sets typically differ in critical parameters like origin, resolution, coordinate reference system, encoding, format, etc. It is important to develop mechanisms that facilitate access to relevant data resources.

The data cube concept has emerged as a solution for organizing a repository of harmonized geospatial raster datasets [1] [2], [3]. In computer technology, a data cube is a multi-dimensional array of values [4]. Those values represent certain facts that can be organized along various aspects. For instance, results of a vote can be considered along political party, voting districts, age or gender of candidates, year of election, etc. These aspects correspond to the axes of the multi-dimensional array and the values live in cells inside this array.

In the geospatial domain, the data cube approach was first used in the Earth Observation (EO) community for organizing vast amounts of satellite images [5]. In this application area time is a very important dimension, as satellite imagery are typically available in extensive time series. The four most typical data cube dimensions in an EO application thus are the two geospatial coordinate axes, time and the imagery type.

In the case of a geospatial data cube, the predominant axes naturally are latitude, longitude and the possible height. A voxel-based approach for organizing truly 3D geodata is also possible [6]. The cell value represents the fact about the physical environment that the data set happens to describe. Thus, content theme can be seen as a predominant axis of a geospatial data cube. The most important benefit of organizing data sets as a data cube repository is immediate availability of the contents for integrated analysis. The approach thus aims at fulfilling the goals of the concept Analysis Ready Data (ARD) [7].

The most important positive aspects that a multidimensional data cube approach provides for geospatial application can be listed as

- A data cube forms an integrated data repository, where harmonized content layers are ready for use without troublesome data preparation steps.
- Ingested datasets are aligned on pixel level and thus can be readily integrated for visualization and analysis.
- A data cube can be accessed along any of its dimensions, allowing for new kinds of knowledge retrieval and spatial analysis processes.
- A joint data cube repository enables data providers to deliver content through a new, easy-to-use channel, thus expanding their user base.
- The simplicity of raster data processing can be more effectively and widely exploited.
- By organizing a data cube as a cloud-based service with Web-friendly APIs, the potential use scenarios can be further widened.

The importance of the data cube concept for geospatial application domain is also demonstrated by the fact that the Open Geospatial Consortium has initiated standardization work on the subject [8]. Another significant development is the Open Data Cube (ODC) initiative. ODC is an open source software library for organizing vast amounts of satellite imagery according to principles of data cube [9]. The software

includes components for cataloguing and indexing EO resources and for ingesting those resources into a harmonized and optimized storage format for easy data retrieval. Furthermore, the ODC library provides tools for accessing and downloading content and for performing analysis operations on it. The ODC community has also developed tools for visual exploration and statistical analysis of the data cube contents. Concrete national and regional data cube implementations based on ODC include the Australian data cube called Digital Earth Australia [10], the Swiss Data Cube [11], Columbian Cube, Vietnam Open Data Cube, and the Africa Regional Data Cube covering Kenya, Senegal, Sierra Leone, Ghana and Tanzania [12].

ODC can also be applied to data resources other than EO data. The examples mentioned in the ODC documentation include gridded data sets like elevation models, geophysical grids and other interpolated surfaces. However, most of the activities around geospatial data cubes focus on satellite image processing, in general, and on their time series applications, in particular [5].

An initiative has been launched in Finland to build a multi-resolution, cloud service-based geodata cube, called GeoCubes Finland, containing some of the most important national geodatasets [13]. The contents of the GeoCubes include data layers like Digital Elevation Model (DEM), administrative areas, land use, surface deposits and various attributes of the national forest inventory. The content layers are provided by the governmental agencies that collect and continuously maintain them. GeoCubes Finland thus differs from other geodata-related data cube implementations by not focusing on satellite imagery, but rather on other traditional geodata sets. The GeoCubes development is being carried out in the context of a large research infrastructure development program in Finland, called Open Geospatial Information Infrastructure for Research (oGIIR) [14]. Datasets to the GeoCubes are being provided by the National Land Survey of Finland (NLS), the Finnish Environment Institute (SYKE), the Geological Survey of Finland (GTK), and the National Resources Institute Finland (LUKE).

In Section II, the basic principles of the GeoCubes Finland are presented. Section III details the technical implementation of the GeoCubes data repository. Data ingestion process is described in Section IV. Section V discusses the custom content access API of the GeoCubes Finland data repository in more detail. Section VI presents the two GeoCubes Finland client platforms developed. In Sections VII and VIII, some visualization related considerations and example applications are discussed. Section IX concludes the paper.

## II. GEOCUBES FINLAND

### A. General

A feature that makes GeoCubes Finland different from other geospatial data cube implementations is its multi-resolution nature. A set of fixed resolution levels (1, 2, 5, 10, 20, 50, 100, 200, 500 and 1000 m) have been selected to store the contained data sets. The resolutions are selected to facilitate processing of analysis operations, typically run on round resolution values, and to enable easy integration with

statistical and other auxiliary data sets. Opposite to resolution levels typical for web mapping schemas, such as the powers of two in Google Maps, in GeoCubes Finland the resolution levels are selected to serve the user in analysis tasks. The available set of resolution levels depend on the original accuracy of the source data set. Thus, the finest resolution of most of the GeoCubes Finland's content layers is 10 m. Only some national data sets can be reasonably be represented in 1 m resolution. These include the most accurate digital elevation model and the administrative division data sets.

Resolution levels could possibly be seen as one dimension of a multi-dimensional geodata cube, but each resolution level has an individual range for the cube's coordinate-related axes. Therefore, the resolution levels of the cube must in fact be seen as a set of separate cube instances. Hence, the plural form of the cube's name: GeoCubes Finland.

In addition to industry-standard access interface to coverage data, the Web Coverage Service (WCS) [15], the GeoCubes contents can be accessed via direct file URLs and through the custom-built GeoCubes API.

### B. Contents

The contents of GeoCubes Finland include a representative selection of spatial data sets maintained by governmental research organisations in Finland (like SYKE, LUKE and GTK). As reference data, some general-purpose data sets provided by the NLS are also included. Data sets are organised as individual layers of information with common representational properties for easy integration and analysis.

Examples of data sets stored in GeoCubes in the first phase include high-resolution elevation models (from the NLS), land-use layers (from the SYKE), superficial deposits layers (from the GTK) and national forest inventory layers (from the LUKE). The latest additions to the GeoCubes Finland data layers are slope and aspect of the terrain, computed from the 2 m resolution DEM, sea regions and the cropland field parcels layer.

### C. Metadata

Metadata concerning the data sets stored in GeoCubes Finland are provided as a centralised resource. Because of the nature of the data sets, special attention is put on providing descriptive information about the classifications applied in raster layers. This information is predominantly made available as internal metadata fields of the raster data file. The access API developed to the GeoCubes supports metadata queries, too. As GeoCubes provides multi-resolution data storage, the applied nomenclature in some cases form hierarchical classification structures.

### D. Data Encoding

The encoding of GeoCubes Finland cell values depends on the nature of the data set being represented. Both classified data sets (like land use or superficial deposits maps) and data sets with continuous value ranges (like Digital Elevation Models (DEMs) or orthophotos) are included. No-data areas are represented as zero-valued cells in classified data sets and by a separate mask channel in data sets with continuous value ranges.

### E. Grid

The standardised grid applied in GeoCubes is based on the Finnish national Coordinate Reference System (CRS) ETRS-TM35FIN (EPSG code 3067). This projected CRS is compatible with the pan-European ETRS89 system. ETRS-TM35FIN covers the whole country in one projection zone and has the false easting value of 500 000 m on its central meridian at 27°E longitude. The origin of the GeoCubes Finland's grid (top-left corner) is located at the coordinate point (0, 7800000). The easting value of the origin is selected to avoid negative coordinates. The northing coordinate value is selected as a round 100 km value, allowing for good coverage of the country.

### III. TECHNICAL IMPLEMENTATION OF THE REPOSITORY

GeoCubes Finland's data storage is implemented as a set of Cloud Optimized GeoTIFF (COG) files [16], [17]. These files are stored on a cloud service platform, organized in directories by data provider, data set and the edition of data set. The area of the country is divided into tessellation of sixty 100 km \* 100 km sized blocks, each maintained as a separate GeoTIFF file to ease processing tasks. The tessellation also facilitates parallelization of various processing steps and works as a rudimentary spatial index. All the files can always be accessed in a straightforward manner via http (HyperText Transfer Protocol) by their URLs (Uniform Resource Locator). The block tessellation is shown in Figure 1.

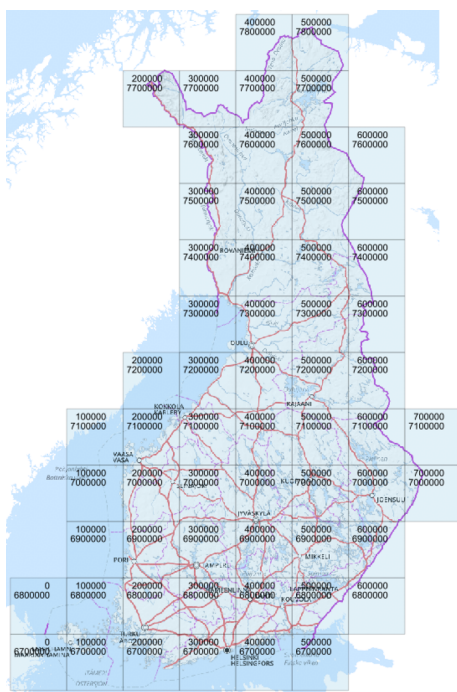


Figure 1. The sixty GeoCubes Finland blocks covering the country and coordinates of their upper-left corner in the national ETRS-TM35FIN coordinate reference system.

A GeoTIFF file arranged according to COG specification contains overviews and has its raster content organized as

tiles. The overviews are ordered in sections from the coarsest to the most detailed, followed by the full resolution raster. The metadata describing the offset of each section must be at the beginning of the file. With this file structure, a calling COG-aware client can make use of the http 'GET Range' query, which enables an indicated subsection of a file to be requested. By first requesting the metadata portion, the client can then select an appropriate range of bytes to only download the needed geospatial area of an appropriate resolution, i.e. overview level. This mechanism significantly improves the efficiency of raster data retrieval for cloud-service based applications. In the rudimentary tests carried out on the GeoCubes Finland COG files, it has been noticed that speed improvements of up to 80 % can be achieved with Cloud Optimized files when compared to traditional GeoTIFF files.

The multiple resolutions are implemented in two ways, both as internal GeoTIFF overview layers and as individual external resolution specific GeoTIFF files. Internal overviews facilitate easy content delivery, as all resolutions are contained inside as single file. Separate resolution specific GeoTIFF files in turn enable better control of resolutions for computations run in external applications. The overview functionality, like most other computing tasks in GeoCubes Finland, is implemented using Geospatial Data Abstraction Library (GDAL) [18]. Python-based processing tasks are performed making use of the Rasterio library [19].

To collect the individual files as a complete view covering the whole country, the so-called Virtual Format mechanism (VRT) of GDAL is used [20]. In this approach, an XML-formatted text file is used to refer to the set of image files that constitute the integrated view. As such, a single VRT file can for instance refer to all external GeoTIFF overview files on certain resolution level, thus facilitating analysis and visualization of larger areas. In the same way, a VRT file can refer to various resolution-specific files in a certain tessellation block, providing a light-weight representation of the multi-resolution data set of that block. VRT files can be easily transported across network connections. Because the file references are stored as absolute addresses, the VRT file can be opened in a third-party application. This approach limits downloading of raster content to the spatial area and to the resolution level the user actually needs.

### IV. DATA INGESTION PROCESSING

In the data ingestion process the data sets being imported to GeoCubes Finland repository are transformed to the common form. Vector data sets are first rasterized. In the rasterization process the value of attribute selected for GeoCubes is stored to the cells. The spatial resolution applied in the rasterization is selected from the list of GeoCubes resolutions, based on the accuracy of the data set in question. All of the data sets currently available in the GeoCubes repository have base resolution of either 1 m or 10 m. In the rasterization process the resulting raster data is stored as GeoTIFF files in the block structure of GeoCubes. If vector data is available in several scales, the smaller scale versions are rasterized and used as the source for overview layers on appropriate resolutions.

If the source data set is available as raster data, the data set is processed to resample the cells to the GeoCubes grid and produce the block-wise GeoTIFF files. In some cases, the classification schemes or data collection resolutions have changed along time between data set editions. This kind of discrepancies must be corrected in the data ingestion process to produce a harmonized collection of repository layers.

The next step in the data ingestion process is to produce the coarser resolution representations as GeoTIFF overview layers. The generalization procedure available in the GDAL's 'BuildOverviews' function is used in the processing of coarser resolutions. An appropriate resampling method is selected, depending on the nature of the source data set. In the process both the resolution levels are produced both as internal GeoTIFF overview layers and as independent single-resolution GeoTIFF files. The multiresolution files are then transformed into the COG form. For categorical data sets the metadata, color table and category information are written to the GeoTIFF files.

In case of categorized data sets, generalization is a delicate process, in which the distribution of categories should be maintained as accurately as possible. If the source data set provides a hierarchical classification scheme, more general classes are used on the coarser resolution levels. In some cases, the generalization procedure has been carried out by the data provider using custom data-specific methods and the resulting multiresolution GeoTIFF files are then used as the input for the GeoCubes data ingestion process.

Next the VRT files are processed. The multiresolution versions of the block-wise GeoTIFF files are combined together in a VRT file. The single resolution files on all available resolution levels are processed in the same way. In addition, for each block a set of VRT files is created, so that for each available resolution level a VRT file is available containing this base resolution and all the coarser resolutions. This arrangement facilitates dynamic provision of multiresolution representations of the content in various base resolutions. Finally, these block-wise VRT files are combined together as VRT files covering the whole country.

Data ingestion procedure is automated by Python script-based processing. GDAL's Python API is used extensively [21]. The main spatial functions like vector data access, rasterization, raster data generalization, georeferencing and GeoTIFF manipulation are based on the use of GDAL Python API. All the processing of raster data arrays is done using Python's NumPy library [22]. The fully automated ingestion process facilitates processing of new data set editions as they become available.

## V. CONTENT ACCESS API

An API has been developed for exploring, downloading and analyzing the GeoCubes Finland contents [23]. The API is designed according to the principles of RESTful Web API [24], in which the request is defined by the path components of the query URL. The API provides requests for querying the basic metadata of the repository, querying cell values on a given location, downloading content by bounding box, polygon or administrative unit and analyzing the content in terms of cell value distribution, change detection, etc. The

analysis operations are run on the server platform, without downloading any content to the client side.

In all content queries, the query can be run on a desired resolution level and year. The multi-resolution structure of the GeoCubes repository gives to the user a flexible choice on speed vs. accuracy of the operation. For instance, an analysis procedure under development can be first tested on a coarse resolution level, and the real, time-consuming run on the finest resolution be carried out only when seemed appropriate. In case of analysis based on visual exploration, the resolution level, on which the analysis is run, can always be matched with the zoom level of the visualization. This way the analysis can be run in roughly constant time and the interactivity level of the application can be kept stable.

The general scheme of the access API can be described as follows:

```
what to do /
  on which resolution level /
    with which content layer /
      where /
        when /
          how
```

As an example, the query for the cell value on a given location becomes as:

```
legend/500/mvmi-paatyyppi/340500,6695000/2009
```

where mvmi-paatyyppi (in Finnish) is one of the themes (forest inventory data) of GeoCubes.

Another query would request download of a content layer inside the given list of municipalities ('kuntajako' in Finnish), from the given resolution level and year:

```
clip/100/corine/kuntajako:734,761,834,433,224,444,927/2000
```

The resulting data set, requested using the GeoCubes Web client, is shown in Figure 2.

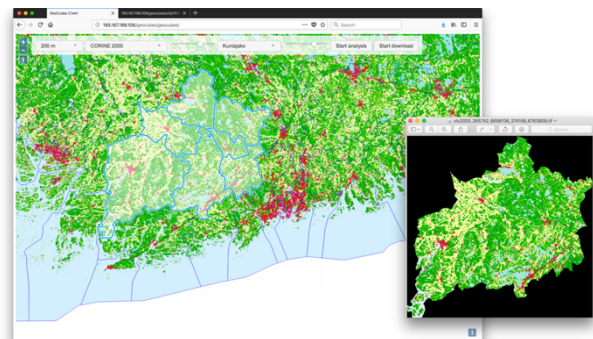


Figure 2. A CORINE data set downloaded via GeoCubes API using administrative units' boundaries as selection area.

Downloading of a DEM from two blocks in VRT format with multiresolution content would go as:

```
clip/20/km10/blocks:300000,6900000,
300000,6800000/2018/vrt:mr
```

Download of a terrain slope layer, computed from the 2 m resolution DEM, in horizontal resolution of 1 m, from inside a user-defined polygon would be initiated like this:

```
clip/1/km2slope/polygon:358978,6668706,358918
,6668645,358923,6668597,358980,6668549,359093
,6668579,359096,6668668,359017,6668719,358978
,6668706/2019
```

An example of an analysis would be change detection between CORINE versions 2000 and 2012 in land use type fields ('pellot' in Finnish) inside the given bounding box:

```
analyse:changedetect/10/corine:pellot
/bbox:225700,6660000,494300,6740000/2000,2012
```

The result of the above analysis is an image depicting the changes in red (removals) and green (additions).

Another analysis would determine the distribution of cell values inside the two indicated counties ('maakuntajako' in Finnish):

```
analyse:distribution/500/mvmi-
maaluokka/maakuntajako:04,06/2015
```

The result is a list of existing cell values together with their frequencies. The analysis results visualized in the GeoCubes Web client using the D3.js library [25] is shown in Figure 3.

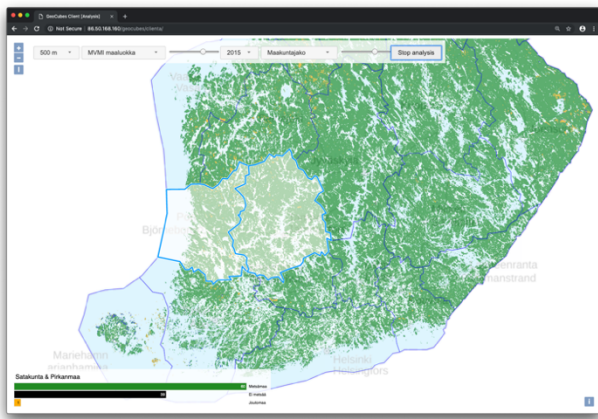


Figure 3. GeoCubes analysis results indicate distribution of data values inside the selected area (shown as a bar chart in the left bottom corner).

As an auxiliary data source for analysis purposes, the Finnish administrative areas in four different levels are also available in vector form in the GeoCubes repository.

The GeoCubes API has been implemented as a Web service using Django Web framework [26], together with Python-based service side scripts making an extensive use of the Python API of GDAL. In addition to the custom API, the most relevant service interfaces standardized by the Open Geospatial Consortium are also supported. The main components of the GeoCubes Finland platform are shown in Figure 4.

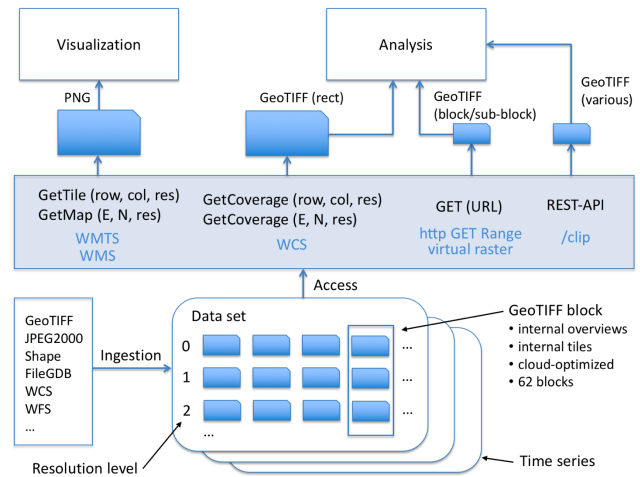


Figure 4. The main system components of the GeoCubes Finland platform.

## VI. CLIENT PLATFORMS

Two main software projects have been accomplished to develop client platforms for supporting the GeoCubes data usage. An Open Layers -based web client has been developed to demonstrate the data contents and access API of the GeoCubes repository. The application provides fast and easy visualization of the GeoCubes content layers, together with interactive querying of the layer cell values. The data retrieval options include download by bounding box, user-defined polygon, administrative area and GeoCubes tessellation block. The user interface of the web client is depicted in Figure 3.

A client application capable of accessing GeoCubes API has also been developed as a QGIS plugin [27]. The plugin provides an exploration tool for interactive visualization and querying of the GeoCubes data layers. Data download can be requested using a bounding box, polygon or administrative unit-based selection. The resolution level can be selected explicitly. The downloaded data set can be directly inserted as a new layer into the current QGIS project, or it can be stored to the local disc for later use. An example of the user interface of the plugin is presented in Figure 5.

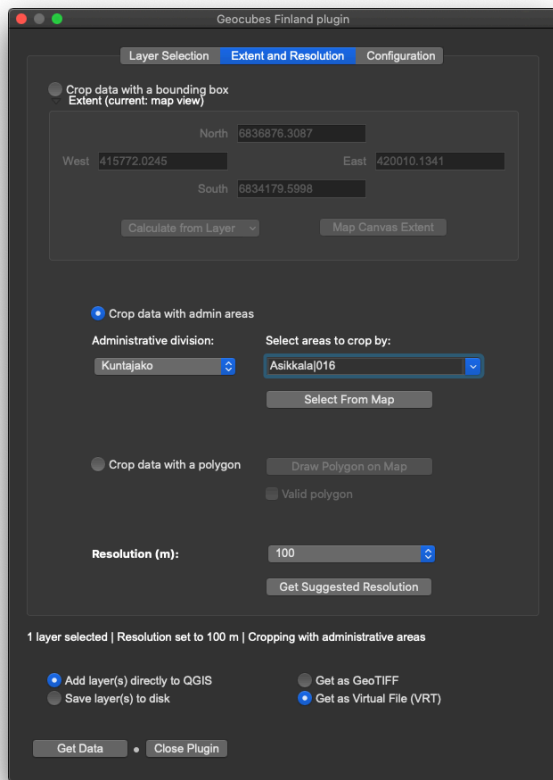


Figure 5. The user interface of the QGIS GeoCubes Finland plugin.

## VII. VISUALIZATION CONSIDERATIONS

For basic visualization of the content layers, a Web Map Tile Service (WMTS) is available on the GeoCubes platform. The service makes use of a Web Map Service (WMS) that uses a VRT file combining the original GeoTIFF files as its source. This arrangement enables both ad hoc visualizations through WMS and cache-based static image delivery via WMTS. In the case of the WMTS service, the pre-rendered tiles are available on the fixed resolution levels of the GeoCubes. For best visual presentation, the client should apply scale levels that correspond to those resolutions. The WMS implementation of GeoCubes platform is based on MapServer and the WMTS service on MapProxy.

Visualization of administrative areas poses a specific challenge. Because areas are represented as raster data to facilitate analysis with other content layers, boundary lines cannot be used to separate an area from the neighboring areas. The only way to reliably display the administrative division of the country in a legible manner is to ensure that neighboring areas are always presented with clearly distinguishable colors. The Finnish lowest level administrative division has more than 300 municipalities. Thus, to achieve an appropriate data set with cell values corresponding to the real municipality codes, a raster with 16-bit cell values was created. However, for Web browser-based visualization, a 256-colour PNG

image is required. A custom software module was developed that creates random color components for a 256-colour RGB color table and applies the colors to municipalities so that the three-color components of a municipality always differ more than 30 units from all of the neighboring areas' color components. The result is a bright and colorful municipality map, in which all areas can be reliably distinguished (Figure 6).

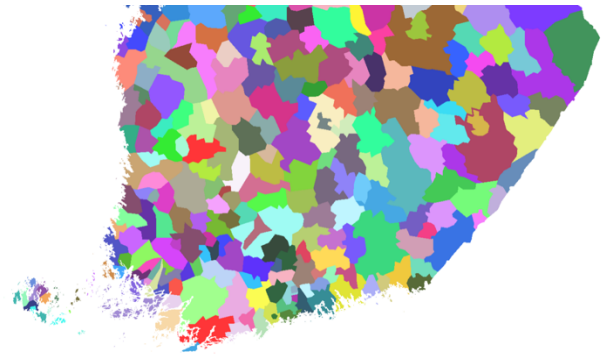


Figure 6. A municipality map with well distinguishable colors.

Another example of enhanced visualization tested in GeoCubes Finland is the locally stretched DEM visualization. The terrain of the country is mostly flat, higher elevations existing only in the northern Lapland. A consistent, stable country-wide DEM visualization thus depicts flat areas with very limited color scale, making it impossible to recognize subtle height differences. A dynamic visualization module was developed for GeoCubes that stretches every individual view requested by the client to full color scale. As the GeoCubes data repository has multiple resolution levels, the visualization processing can be performed efficiently in constant time, independently of the presentation scale of the client. The two images in Figure 7 depict the difference between fixed and dynamic DEM visualization on a particularly flat area in the Finnish Ostrobothnia.

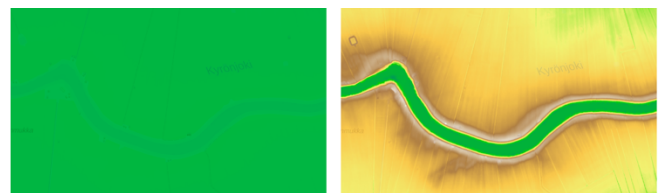


Figure 7. A static DEM visualization compared with a dynamic, locally stretched graphic scale.

A 3D example of visualization of GeoCubes' contents is depicted in Figure 8. Here, DEM data is extracted from the GeoCubes data repository around possible crater locations, where a meteorite impact is deemed to be behind the peculiar rounded land form. The multiresolution contents of the GeoCubes repository supports effective 3D visualization of crater areas of various sizes.

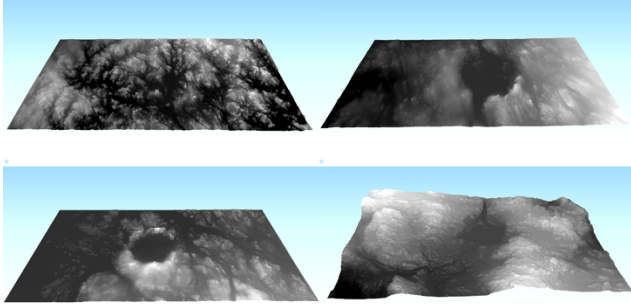


Figure 8. Possible meteorite impact craters visualized, based on GeoCubes DEM. Big scale differences in visualizations are supported by GeoCubes' multiresolution contents.

## VIII. APPLICATION EXAMPLES

The analysis processes can be effectively run on different resolution levels and their results are readily available for visual exploration over the whole range of scales supported by the geodatacube's resolutions. The GeoCubes API has been designed to support access, querying and analysis of the geodatacube's contents on the resolution level appropriate for the actual use situation.

### A. Field Inspection

As an example of using GeoCubes API, a field inspection of repository cell values has been tested. In this approach, the inspector moves around on the terrain and constantly receives values of the GeoCubes layer of interest into his cell phone. This way, the inspector can compare the environment around him with the stored category values, for instance, he can determine the correctness of forest type information – and do that on all available resolution levels of the repository.

In the developed pilot implementation, the position of the inspector is recorded with an application called Owntracks. The location is reported back to a server (Eclipse Mosquitto) [28] implementing the ISO-standardized Message Queuing Telemetry Transport (MQTT) -protocol. The user is presented with a map-based application (Leaflet with Realtime extension). When the map application gets notice from the MQTT server that the user has moved to a new location, it then send a request to the 'legend' operation of the GeoCubes API together with the location information. As a result, it will get cell values from the requested layer on all available resolution levels and can compare them with the reality around him. The pilot's architecture is presented in Figure 9.

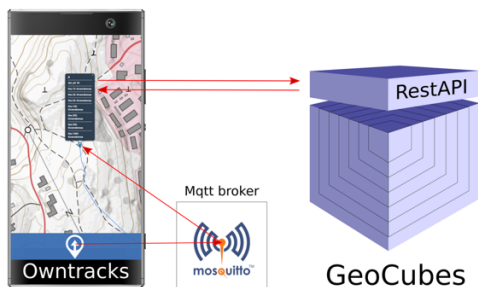


Figure 9. A field inspector accessing GeoCubes cell values using Mosquitto server.

### B. Route Finding

In another analysis example, the GeoCubes DEM layer is used for route finding for a forest vehicle that has certain limits for movement in steep slopes. DEM data for study area was acquired by accessing the GeoCubes API's 'clip' operation from within the QGIS application [29]. Then, analysis functions available in QGIS were used to compute slope values for the area and these were used as the cost surface for route finding. Areas too steep for the vehicle were excluded from the computation. The case study demonstrates the use of the GeoCubes' content for analysis using tools outside the GeoCubes platform. The same analysis could also be run in different resolution levels, depending on the needs of the application. The resulting route alternatives are depicted in Figure 10 using QGIS plugin [30] for three.js, a JavaScript library for 3D visualization [31].

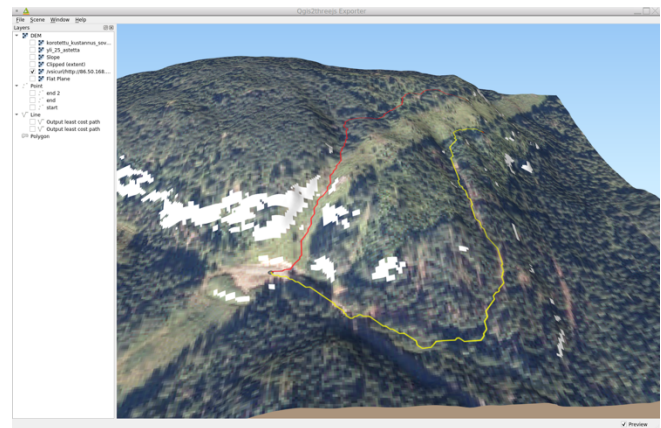


Figure 10. Route finding in QGIS with DEM from GeoCubes using slope as cost surface. White areas denote terrain that has been classified to be too steep and is excluded. The red and yellow lines denote found alternative routes.

### C. Wind Exposure Analysis

Another GeoCubes data application example involves creating a forecast map of perceived temperature, based on location-dependent exposure to wind. The result is a "winter map" with a background layer, adapted to reflect the snowy conditions of the Nordic winter landscape, together with an overlay indicating the direction of wind and the level of wind chill-affected perceived temperature. The approach makes use of the GeoCubes-provided cloud-optimized multiresolution DEM data, combined with the weather forecast information accessed online from the Finnish Meteorological Institute (FMI).

The service is implemented as a PHP: Hypertext Preprocessor script making an extensive use of GDAL's command line interface. At the core of the service is the GDAL command 'gdal\_translate', which accesses the online VRT files of the GeoCubes repository by direct URL reference. The VRT file provides a convenient single point access to all sixty multiresolution GOC files that cover the whole country. Using the 'projwin' and 'outsize' parameters of the 'gdal\_translate' command, one can specify from which area the data is to be retrieved and what is the desired size of

the resulting image. These parameters together determine the resolution level that is accessed in the GeoCubes multiresolution data storage. This arrangement facilitates an easy and efficient presentation of the analysis results in various scales, depending on the zooming levels selected in the user interface.

Once the request for a “winter map” comes to the service, the PHP script will access the FMI’s weather forecast service through a Web Feature Service (WFS) interface, and then parses the required wind direction information from the response. Consequently, the script will request the needed DEM data from the GeoCubes service. The DEM is processed to produce the slope and aspect layers, needed for computing the perceived temperature. After the computation, the resulting wind chill-affected perceived temperature map is rendered. The orthophoto layer is requested from GeoCubes and dynamically tuned to reflect the snowy winter conditions by applying appropriate image filtering tools. Finally, these two map layers are combined and the resulting “winter map” is returned to the client application. An exemplary map, depicting the area of the Levi ski resort in the Northern Finland and visualized for 3D viewing in a three.js-based web client, is shown in Figure 11.

## IX. CONCLUSIONS

One of the major obstacles for wider use of geospatial raster data sets in different research and analysis scenarios is the work required for pre-processing the available data sets. This often involves coordinate reference system transformations, resampling procedures, coding system translations, integration of map sheet-based data files, etc. To facilitate the introduction of geospatial data sets into research processes in a multidisciplinary setting, a harmonized, easy-to-access data storage would be really beneficial. In the GeoCubes Finland initiative, this kind of approach has been taken.

A representative set of geospatial data sets with national coverage has been ingested into the cloud service-based data repository. The pre-processing phase involves operations like rasterizing vector-formatted source data sets, resampling of source data into the set of standardized GeoCubes resolution levels, harmonization of value coding systems between different data set editions, encoding of the resulting raster content into the common cloud-optimized image format and storing it to the cloud repository, both as binary images files and as textual VRT representations.

By storing the raster data sets in multiple resolution levels, the GeoCubes repository can support certain use cases very efficiently. These include interactive visual exploration of on-the-fly analysis results and testing of an analysis procedure on coarse resolution levels before launching an accurate analysis on detailed levels. The result layer of an analysis can be configured as a new GeoCubes content layer and thus be run dynamically by the calling visualization client. The analysis procedure will always select the resolution level closest to the visual scale, thus enabling nearly constant processing times to be maintained.

Further work on the GeoCubes Finland platform will focus on adding new layers to the repository’s data collection, developing modular analysis functions for server-side processing, and enhancing the GeoCubes client applications. More user testing will be carried out to gather feedback and guidance for further development of the platform.

## ACKNOWLEDGMENT

We made use of computing resources provided by the Open Geospatial Information Infrastructure for Research (oGIIR, urn:nbn:fi:research-infras-2016072513) funded by the Academy of Finland and CSC – The IT Center for Science Ltd.

## REFERENCES

- [1] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, “Harmonized Multiresolution Geodata Cube for Efficient Raster Data Analysis and Visualization,” in *GEOProcessing 2020: The Twelfth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain*, Nov 21-25, 2020, pp. 36–41, Accessed: Dec. 7, 2020. [Online]. Available: [https://www.thinkmind.org/index.php?view=article&articleid=geoprocessing\\_2020\\_1\\_60\\_30055](https://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2020_1_60_30055).
- [2] P. Baumann, “The Datacube Manifesto,” 2017, Accessed Dec. 7, 2020. [Online]. Available: [https://external.opengeospatial.org/twiki\\_public/pub/CoveragesDWG/Datacubes/The-Datacube-Manifesto.pdf](https://external.opengeospatial.org/twiki_public/pub/CoveragesDWG/Datacubes/The-Datacube-Manifesto.pdf).
- [3] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, “GeoCubes Finland - A Unified Approach for Managing Multi-resolution Raster Geodata in a National Geospatial Research Infrastructure,” in *GEOProcessing 2018: The Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services*, Mar. 2018, pp. 18–22, Accessed: Dec. 7, 2020. [Online]. Available: [http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing\\_2018\\_2\\_10\\_30069](http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2018_2_10_30069).
- [4] Wikipedia, “Data cube,” 2020. Accessed Dec. 7, 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Data\\_cube](https://en.wikipedia.org/wiki/Data_cube)
- [5] A. Lewis *et al.*, “Rapid, high-resolution detection of environmental change over continental scales from satellite data – the Earth Observation Data Cube,” *International Journal of Digital Earth*, vol. 9, no. 1, pp. 106–111, Jan. 2016, doi: 10.1080/17538947.2015.1111952.
- [6] U. Pyysalo and T. Sarjakoski, “Voxel Approach to Landscape Modelling,” in *The International Archives of the Photogrammetry and Remote Sensing*, 2008, pp. 563–568, Accessed: Dec. 7, 2020. [Online]. Available: <https://www.semanticscholar.org/paper/Voxel-Approach-to-Landscape-Modelling-Pyysalo/a2074c8487c028670348c6efa051d6590a41e6c8>.
- [7] G. Giuliani *et al.*, “Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD),” *Big Earth Data*, vol. 1, no. 1–2, pp. 100–117, Dec. 2017, doi: 10.1080/20964471.2017.1398903.
- [8] P. Baumann, D. Misev, P. Hogan, R. Brackin, K. Thayer, and E. Hirschorn, “Datacube Domain Working Group Carter,” 2017. Accessed Dec. 7, 2020. [Online]. Available:



- [https://external.openeospatial.org/twiki\\_public/pub/CoveragesDWG/Datacubes/17-071\\_Datacube-DWG\\_Charter.pdf](https://external.openeospatial.org/twiki_public/pub/CoveragesDWG/Datacubes/17-071_Datacube-DWG_Charter.pdf)
- [9] ODC, “Open Data Cube,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://www.opendatacube.org/>
- [10] A. Lewis *et al.*, “The Australian Geoscience Data Cube — Foundations and lessons learned,” *Remote Sensing of Environment*, vol. 202, pp. 276–292, Dec. 2017, doi: 10.1016/j.rse.2017.03.015.
- [11] SDC, “Swiss Data Cube (SDC) – EO for monitoring the environment of Switzerland in space and time,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://www.swissdatacube.org/>
- [12] Global Partnership for Sustainable Development Data, “Africa Regional Data Cube,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <http://www.data4sdgs.org/index.php/initiatives/africa-regional-data-cube>
- [13] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, “Supporting wide user-base in raster analysis - GeoCubes Finland,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2018, vol. 42, no. 4, pp. 397–402, doi: 10.5194/isprs-archives-XLII-4-329-2018.
- [14] OGIIR, “Open Geospatial Information Infrastructure for Research,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <http://www.geoportti.fi/about/>
- [15] OGC, “Web Coverage Service,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://www.ogc.org/standards/wcs>
- [16] OSGeo, “GeoTIFF,” 2019. Accessed Dec. 7, 2020. [Online]. Available: <https://trac.osgeo.org/geotiff/>
- [17] COG, “Cloud Optimized GeoTIFF,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://www.cogeo.org/>
- [18] GDAL, “Geospatial Data Abstraction Library,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://gdal.org/>
- [19] MapBox, “Rasterio: access to geospatial raster data — rasterio documentation,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://rasterio.readthedocs.io/en/latest/>
- [20] GDAL, “VRT – GDAL Virtual Format,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://gdal.org/drivers/raster/vrt.html>
- [21] GDAL, “GDAL Python API,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://gdal.org/python/>
- [22] NumPy, “NumPy - The fundamental package for scientific computing with Python,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://numpy.org/>
- [23] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, “Flexible Access to a Harmonised Multi-resolution Raster Geodata Storage in the Cloud,” in *GEOProcessing 2019: The Eleventh International Conference on Advanced Geographic Information Systems, Applications, and Services*, Feb. 2019, pp. 26–28, Accessed: Dec. 7, 2020. [Online]. Available: [http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing\\_2019\\_2\\_20\\_30054](http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2019_2_20_30054).
- [24] M. Rouse, “What is a RESTful API (REST API) and How Does it Work?,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://searcharchitecture.techtarget.com/definition/RESTful-API>
- [25] M. Bostock, “D3.js - Data-Driven Documents,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://d3js.org/>
- [26] Django Software Foundation, “The Web framework for perfectionists with deadlines | Django,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://www.djangoproject.com/>
- [27] QGIS, “QGIS Plugins,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://plugins.qgis.org/>
- [28] R. Light, “Mosquito: server and client implementation of the MQTT protocol,” *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017, doi: 10.21105/joss.00265.
- [29] QGIS, “QGIS - A Free and Open Source Geographic Information System,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://qgis.org/en/site/>
- [30] QGIS, “Qgis2threejs Plugin Document — Qgis2threejs Plugin 2.4 documentation,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://qgis2threejs.readthedocs.io/en/docs/index.html>
- [31] threejs, “three.js – JavaScript 3D library,” 2020. Accessed Dec. 7, 2020. [Online]. Available: <https://threejs.org/>

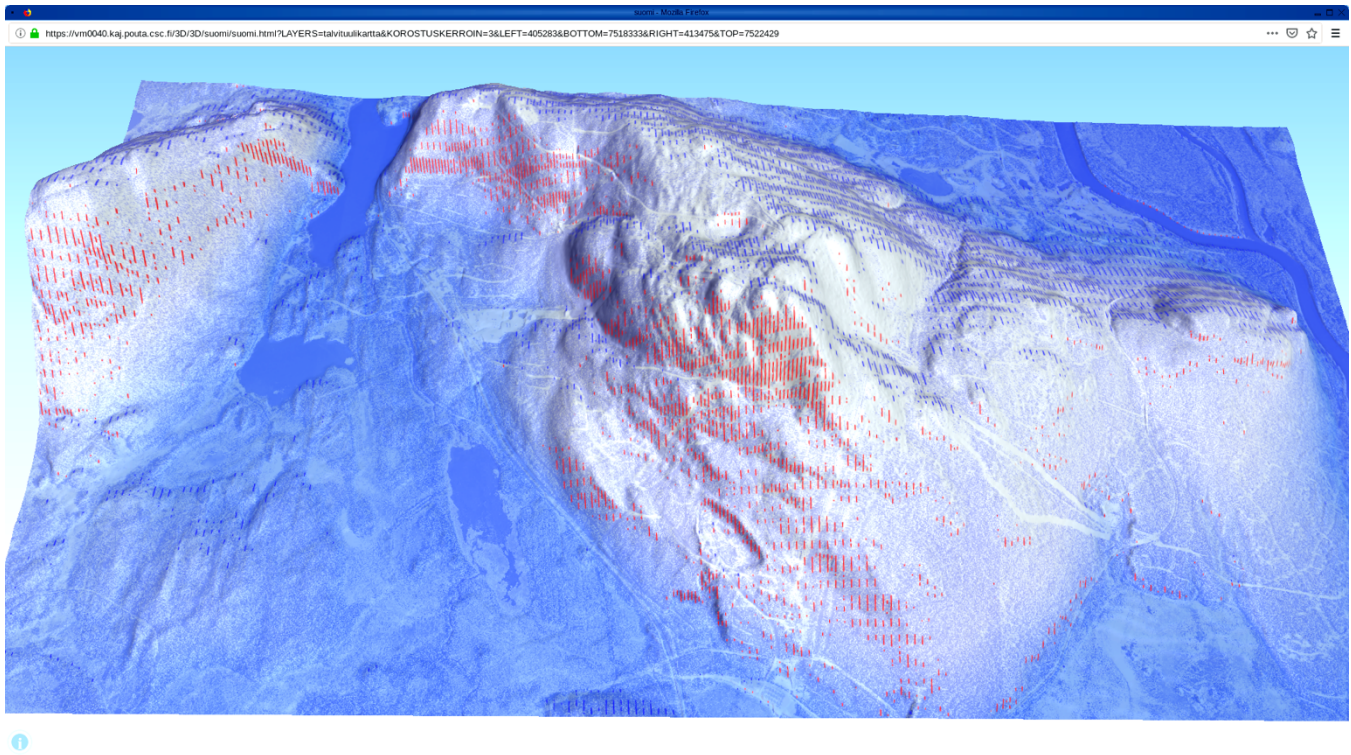


Figure 11. A perceived temperature “winter map” presented for 3D viewing in a three.js-based web client. The areas with red stripes are on the shelter side of the hill, whereas the areas with blue stripes represent chilly slopes, exposed to wind. The orientation of the stripes indicates the direction of the wind and their density the level of the shelter/exposure effect.