A Technical Framework for Visualizing Spatio-temporal Quality Metrics of Volunteered Geographic Information

Oliver ROICK[a,1], Lukas LOOS[a] and Alexander ZIPF[a]

[a] *Chair of GIScience, Heidelberg University, Heidelberg, Germany*

Abstract. The following paper presents a technical framework that allows for visualization of spatio-temporal quality metrics (user activity, topicality or number of features) of Volunteered Geographic Information using the example of OpenStreetMap (OSM). The aim of the system is to allow for analyzing the spatially heterogeneous data quality of VGI by a.) providing a spatio-temporal database of indicators derived from VGI that can be used in more complex statistical analysis and data mining of the data and b.) to provide a web-based visualization for first visual analysis of the data. The framework is based on a database applying a star schema separating geo-spatial and temporal information from attribute values. Further, a web-based client has been implemented on top of a Web Map Service, which allows for visual exploration of the data.

Keywords. Volunteered Geographic Information, Visualization, OpenStreetMap

Introduction

Volunteered Geographic Information (VGI) – geographic data, which is primarily acquired through voluntary effort by private citizens [1] – has become a serious alternative to governmental and commercial geographic data. Several services that are running upon OpenStreetMap data – VGI's most prominent representative – have evidently demonstrated the potential of VGI for real world applications [2,3,4,5,6].

However, several studies have shown that data quality of *Volunteered Geographic Information* varies largely depending on the area being investigated [7,8,9,10,11]. Therefore, quality control is a crucial part when dealing with VGI and applying it to projects. Due to the large size and complex nature of geo-spatial data visual exploration has been proven to be a suitable methodology to reveal patterns and anomalies and therefore quickly assess the quality of the data at a first glance [12,13].

In 2011 we presented OSMatrix[2] – a web-based approach to visually explore quality metrics of OpenstreetMap by portraying several attributes on user activity, topicality of data as well as the occurrence and distribution of certain feature types throughout Europe [14]. OSMatrix originated from the idea that an extensive database of quality related indices may be the foundation for further analysis and data mining [15], so the visualization framework proposed here is just one application scenario.

However, first evaluations of the resulting maps show that the approach reproduces already known insights, such as the data imports in the Netherlands and Estonia or the superior completeness of the street network data in Germany compared to other countries such as Spain or Italy. The brief analysis further revealed small-scale anomalies that turned out to result from local data imports, digitizing from satellite imagery and locally bounded mapping activities. Hence, the approach was proven to be suitable for providing a quick visual overview on certain quality characteristics.

However, in order to gain additional insights on data quality it is required to evaluate the data set over time to learn how certain quality metrics evolve and to identify activity hot spots and areas with stagnating process.

Therefore, the approach presented before [14] has been extended to support processing, storage and management of spatio-temporal quality metrics as well as visualizing the data to support evaluation of the temporal evolution of the data set. The following paper presents the technical framework that allows for the realization of the objective stated above and further, a detailed discussion regarding data processing, storage and visualization is provided.

1. Related Work

Three similar approches have been proposed by Trame & Kessler [16] and van Exel [17,18].

Trame & Kessler [16] visualize version numbers of Point of Interests in the OpenStreetMap data set using a heat map and therefore allow for a visual exploration of regions with high editing activity. Further, tools are provided, which enable researchers to explore editing activities by over time.

In similar fashion, van Exel [17] presents an approach to exploit hot spots of user activity by visualizing version numbers of any node in the OSM database using contour lines. Van Exel further presents a way to portray version numbers and dates of the most recent edit of linear OSM features in a combined manner [18]. The resulting evaluation allows for identification of abandoned features, which should be subject to quality control.

However, the method proposed here differs form the approaches discussed above in way that it attempts to provide a combined overview of several attributes. Besides the version number and time of recent updates OSMatrix provides a multitude of additional data ranging from number of user contributions and number of contributions per user over version numbers and edit times to information on the number of mapped features and attributes as well as the area covered by certain feature types. For a comprehensive overview on the attributes provided please refer to Table 1.

Table 1. Overview of the attributes provided by OSMatrix.

Minimum, Maximum, Average	Cumulated	Area
Number of attributes per feature	Number of attributes	Buildings
Number of contributions per user	Number of features	Allotments
	Number of contributing users	Farmyards/farmland
Version number		Industrial, commercial and residential landuse

2. OSMatrix workflow

Figure 1 depicts a simplified workflow of the processing and visualization architecture of OSMatrix.

For each time step an extract of the OSM data set is imported into the OSMatrix data base and processed as follows. For each hexagonal cell in the matrix the attributes mentioned in Table 1 are calculated and written back to the database. Then cartographic representation of the data is adjusted according to the updated attribute values. The updated visualization rules are encoded in Styled Layer Descriptor (SLD) and Symbology Encoding (SE) and written to the Map Server. Using the client a user is able to request maps and additional temporal information. The information is retrieved from the server through Web Map Service (WMS) and Web Feature Service (WFS) interfaces. The Map service runs on top of the OSMatrix database and therefore pulls the requested data from that database.

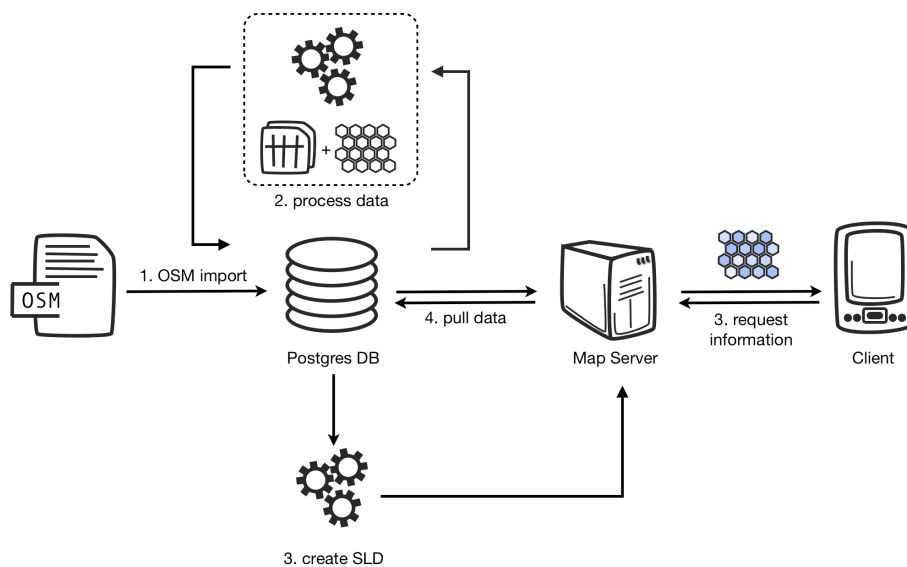In the following sections each component of the system will be discussed in detail.

Figure 1: OSMatrix processing and visualization workflow.

3. Database design

In order to illustrate the spatial distribution of OSM data for Europe the area is divided into hexagonal cells using the open source library GeoTools. For each cell a given number of attributes is then calculated by intersecting the cell with the OSM data.

For storage and further analysis we use a star-schema [19] PostgreSQL/PostGIS database. It is centered around a fact-table containing the calculated attribute values and foreign keys for the dimensional tables cells and attribute type respectively. Two keys pointing to the time dimension are used to provide information about the lifetime

of an attribute value. Improvements to query performance are achieved through a trigger based partitioning of the fact-table.

For our application the multidimensional model is beneficiary since the approach is focused on analytical queries on temporal slices. Updates are performed as batch jobs on a monthly basis. Real time transactions with inserts, updates or deletes do not occur.
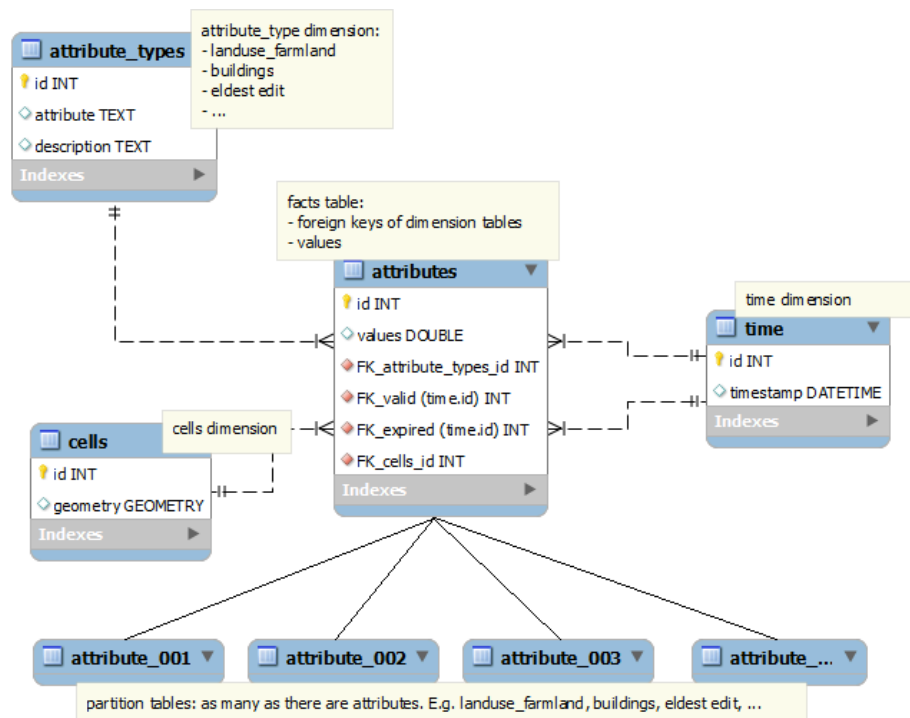
Figure 2: OSMatrix database schema.

The following code demonstrates an exemplary star join. In this case the partition tables are not taken into account.

```
select
  attr.id, cells.id as cell, attr_t.attribute, attr.value,
    t1.time as valid, t2.time as expired
  from attributes as attr
  left join cells on (attr.cell_id = cells.id)
  left join attribute_types as attr_t
    on (attr.attribute_type_id = attr_t.id)
  left join time as t1 on (attr.valid = t1.id)
  left join time as t2 on (attr.expired = t2.id);
```

4. Processing

4.1. Data processing

The data processing is done with Java on a monthly basis. In a first step the OSM dataset for Europe is downloaded from GEOFABRIK[3] and extracted with the osm2pgsql tool to the database. In a next step a comparison between the new dataset and the existing data in the database is carried out and a new timestamp is added to the time table. Changed and new data is loaded into the database and the associated keys the time table are stored or updated. The performance of the process correlates with the amount of attributes to calculate, in particular attribute calculations that imply polygon intersections and clipping.

4.2. Map Styling

On top of the Postgres database an instance of Geoserver is employed to visualize the data. Therefore, an additional step is necessary, that adapts the cartography (i.e. the classification ranges of the choropleth maps) to the new data stored in the database. This algorithm it triggered automatically after the database was updated successfully.

The algorithm first retrieves the value range for the given attribute along with the number of occurrences for each value. Then, the class ranges are calculated based on the occurrence of the values. Therefore, a modified quantile classification is employed. Analysis of the attribute tables have shown that lower attribute values often occur multitudinous in a way that the number of a single value exceeds the size of a single class. The classification process has been adjusted in a way that these values are classified in one class and the size of the following classes is reduced. This further improves the classification for the following values allowing for a finer resolution of the classes.

The classification along with the map styling is then encoded into OGC compatible Styled Layer Descriptor documents using Symbology Encoding to describe the map styling. To dynamically update the map styling Geoserver's REST API[4] is employed which provides methods for retrieving and updating style encodings using HTTP.

5. Client implementation

The corresponding client application (Figure 3) has been implemented using the open source web mapping framework OpenLayers[5] on top of the aforementioned server side components. For each attribute a single layer is available on server side and included for display in the client. In order to support the portrayal of temporal information the application employs Geoserver's application specific WMS parameter viewparams[6], which allows for parameterization of WMS requests on Geoserver's SQL layers. This enables a client to request filtered WMS outputs similar to filters in WFS requests (see example below).

---

3 http://download.geofrabrik.de
4 http://docs.geoserver.org/2.1.3/user/restconfig/rest-config-api.html
5 http://openlayer.org
6 http://docs.geoserver.org/latest/en/user/data/sqlview.html

```
http://osmatrix.geoserver/wms/osmatrix/?
  request=getLayer&
  layer=userContrib&
  viewparams=time:1296758206
  [...]
```

Additionally, it shows histogram statistics for selected cells that illustrate the development of attributes in a certain area during time and provide insights on the evolution of the data set in a given area. The graphs are produced using the Protovis toolkit [20], which employs HTML5 and SVG to render graphics on the fly. The respective data is retrieved using the WFS GetFeatureInfo request using the ID of the given cell as a filter. The server response includes several features, each representing a point in time, which are then evaluated and displayed in a graph.
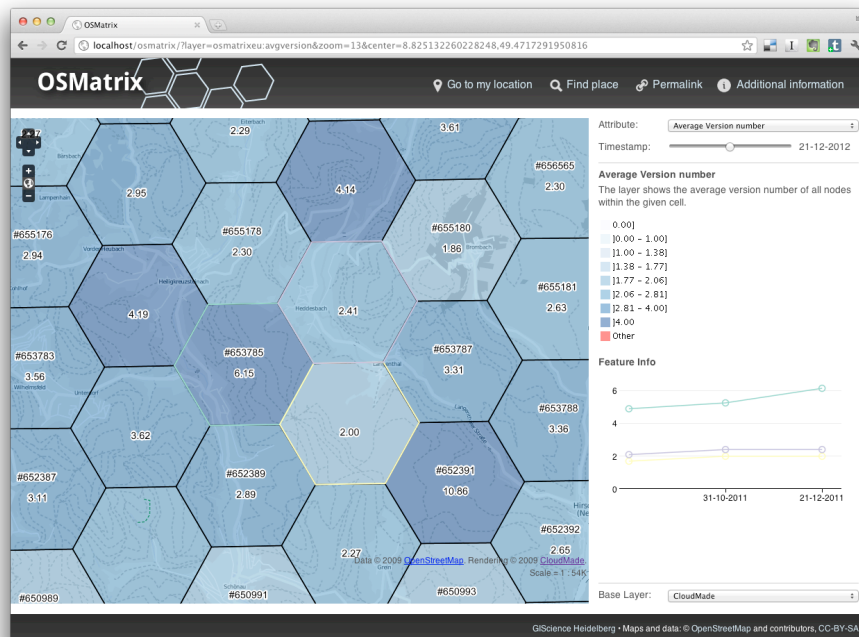

Figure 3: OSMatrix client interface.

6. Future Work

The main future objective is to employ the presented application and derive insights on the temporal evolution and spatial distribution of user activities, completeness and topicality of the data. Additionally it is planned to relate our findings to other data, such as population density or socio-economic parameters in order to improve interpretation of our findings.

Further, improvements to the overall performance of the system are desired. Currently, we face issues when displaying maps at a small scale due to bottlenecks while rendering large amounts of features at once. Possible solutions might be the use of TileCache services or switching from displaying vector features to using raster images at large scales.

References

[1] M. Goodchild. Citizens as sensors: The world of volunteered geography. *GeoJournal*, **69**:211-221, 2007.

[2] M. Goetz, M. Auer, J. Lauer, and A. Zipf. An algorithm based methodology for the creation of a regularly updated global online map derived from Volunteered Geographic Information. *The Fourth International Conference on Advanced Geographic Information Systems, Applications and Services. GEOProcessing 2012*, Valencia, Spain, 2012.

[3] A. Müller, P. Neis, and A. Zipf. Ein Routenplaner für Rollstuhlfahrer auf der Basis von OpenStreetMap-Daten. Konzeption, Realisierung und Perspektiven. *AGIT 2010. Symposium für Angewandte Geoinformatik*, Salzburg, Austria, 2010.

[4] P. Neis and A Zipf. Openrouteservice.org is three times "open": Combining open-source, OpenLS and OpenStreetMap. *GIS Research UK (GISRUK 08)*, Manchester, United Kingdom, 2008.

[5] S. Neubauer, M. Over, A. Schilling, and A. Zipf. Virtual cities 2.0: Generating web-based 3D city models and landscapes based on free and User Generated Data (OpenStreetMap). *GeoViz 2009. Contribution of Geovisualization to the concept of the Digital City*. Workshop, Hamburg, Germany, 2009.

[6] A. Schilling, M. Over, S. Neubauer, P. Neis, G. Walenciak, and A. Zipf. Interoperable location based services for 3D cities on the web using User Generated Content from OpenStreetMap. *UDMS 2009. 27th Urban Data Management Symposium,* Ljubljana, Slovenia, 2009.

[7] J.-F. Girres and G. Touya. Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, **14**(4):435-459, 2010.

[8] M. Haklay. How good is Volunteered Geographical Information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design*, **37**:682-703, 2010.

[9] I. Ludwig, A. Voss, and M. Krause-Traudes. A comparison of the street networks of Navteq and OSM in Germany. In S. Geertman, W. Reinhardt, and F. Toppen, editors, *Advancing Geoinformation Science for a Changing World*, Lecture Notes in Geoinformation and Cartography, 65-84. Springer, 2011.

[10] D. Zielstra and A. Zipf. A comparative study of proprietary geodata and Volunteered Geographic Information for Germany. *AGILE 2010. The 13th AGILE International Conference on Geographic Information Science*, Guimaraes, Portugal, 2010.

[11] P. Neis, D. Zielstra and A. Zipf. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, **4**(1):1-21, 2012.

[12] D. DiBiase. Visualization in the earth sciences. *Earth and Mineral Sciences Bulletin*, **59**(2):13-18, 1990.

[13] J. Dykes, A.M. MacEchren, and M.-J. Kraak. Exploring geovisualization. In J. Dykes, A.M. MacEchren, and M.-J. Kraak, editors, *Exploring geovisualization*, 3-19. Elsevier, 2005.

[14] O. Roick, J. Hagenauer, and A. Zipf. OSMatrix - Grid based analysis and visualization of OpenStreetMap. *State of the Map EU 2011*, Vienna, Austria, 2011.

[15] J. Hagenauer and M. Helbich. Mining urban land-use patterns from volunteered geographic information by means of genetic algorithms and artificial neural networks. *International Journal of Geographical Information Science*, 2011. DOI:10.1080/13658816.2011.619501.

[16] J. Trame and C. Keßler. Exploring the linage of Volunteered Geographic Information with heat maps. *GeoViz*, Hamburg, Germany, 2011.

[17] M. van Exel. A new OpenStreetmap Visualization: Version contour lines, 2011. URL: http://wp.me/pcUHq-5f.

[18] M. van Exel. A new OpenStreetmap Visualization: Taking the Temperature of local OpenStreetMap-Communities, 2011. URL: http://wp.me/pcUHq-6r.

[19] A. Bauer, H. Günzel. Data *Warehouse Systeme. Architektur, Entwicklung, Anwendung,* 217-220, 2009.

[20] M. Bostock and J. Heer. Protovis: A Graphical Toolkit for Visualization *IEEE Transactions on Visualization and Computer Graphics*, **15**(6):1121−1128, 2009.