Accessing the history of objects in OpenStreetMap

Peter Mooney and Padraig Corcoran

Geotechnologies Research Group, Department of Computer Science, National University of Ireland Maynooth (NUIM), Co. Kildare. Ireland +353 (1) 708 3847
Email{peter.mooney, padraig.corcoran}@nuim.ie

1. INTRODUCTION

OpenStreetMap (OSM) is probably the best known example of Volunteered Gegographic Information (VGI). OSM data can be downloaded in OSM-XML format and Shapefile (SHP) format from services such as GeoFabrik (GeoFabrik, 2010). These data can be downloaded in convenient packages such by as country, continent, and in some cases administrative regions. An important characteristic of these downloads is that the OSM spatial data contained within the OSM-XML files or SHP files are close to real-time representations of the spatial data stored in the global OpenStreetMap database. GeoFabrik state that “essentially any change made to the global OSM database is usually reflected in the data packages available for download within 24 hours” (GeoFabrik, 2010). Within these download packages users have access to the most recent version Vt of any spatial object O, in the OSM database. Suppose one wanted to perform a comparison between other versions of O such as comparing the current version Vt and the previous version Vt−1 or indeed all versions V0 , V1 . . . Vt. This is not possible using these downloaded packages. Analysis of OSM is hindered by the quickly growing size of OSM data download packages are and this is a problem for end users. For example, the uncompressed version of OSM for Germany is a 5GB XML file. Processing this file, in it's raw format, is not possible for most users and indeed most desktop machines. This work is motivated by the exciting research potential offered by the analysis of the history of edits to geographical objects in OSM which subsequently could provide many interesting insights into VGI.

The ability to access the spatial and editing history of objects in the OSM database could potentially
lead to some very interesting research questions. These questions include:

• How does an object structurally change over time? How does the representation of the object change over time? (Related Work: spatial representation of geographical features in OSM by Mooney et al. (2010a,b)).
• How does the metadata (tags) associated with an object change over time and can they provide information about the spatial literacy of contributors? (Related Work: Comber et al. (2005))
• Is a given region R of OSM topologically consistent over it’s lifetime or over some time period t1 to t2? (Related Work: Topological consistency of vector data by Corcoran et al. (2011))
• Is a given object O valid over some time period t1 to t2 ? Are changes in the object O reflected in other objects U within the same neighbourhood or region? (Related Work: Quantifying Map Information by Harrie and Stigmar (2010))
• Which OSM contributors have edited a given object O or a set of objects and what is the nature of their contributions? (Related work: how OSM grows/changes are more users contribute in an area by Haklay et al. (2010))

2. DESCRIPTION OF ALGORITHM

In this section we describe our algorithm in detail. Algorithm 1 outlines the steps involved. The software implementation is designed for desktop use only. The coordinates of a bounding rectangle representing a region R is OSM are provided in a plain text configuration file. The software then automatically downloads the OSM-XML for R directly from the OSM Application Programming Interface (API). The feature identification numbers (OSM-ID) are extracted for every polygon and polyline in the OSM XML file. For each object P the history of edits is downloaded as an OSM-XML file from the OSM API. The history file for the object P is then processed. Suppose that the object P has n versions where i = 0 is the first version and i = n − 1 is the final or current version. Then each version Pi of P is stored as the tuple Pi = (ui , Ni , τi , N SRi , G(i), A(i), L(i), Di , T (i)) where the elements of the tuple Pi are as defined as follows: ui is the user id of the OSM contributor who edited version i ; Ni is the number of nodes in polygon Pi ; τi is the timestamp for the edit ; N SRi is the number of nodes which "survived" from the previous version Pi−1 of polygon Pi ; G(i) is the geometry of Pi ; A(i) is the area of G(i) in hectares (only calculated for polygons) ; L(i) is the length or perimeter of G(i) in meters ; Di is the mean spacing in meters between the adjacent nodes of Pi ; T (i) is the set of tags (keys,values) assigned to this version of Pi which are stored as a comma separated list. Finally, if specified, for each version Pi of the object P a vector data file representation is written out to disk. There are a number of possible output formats: ESRI Shapefile, KML file, or GPX format.

```
Data: OSM Input Region R - bounding rectangle coordinates in (longitude, latitude)
Result: Database table containing edit history of all objects in R

if Area(R) ≤ A_max then
    Download OSM XML;
    Extract IDs of polygons and polylines and POI from the OSM-XML;
    for i ← 0 to n do
        Download history file for object i from the OSM API;
        Process history file for object i and store result in database;
        for v ← 0 to v(i) do
            Store object as a geometry;
            Store all tags associated with the object;
            Store timestamp;
            Store user information;
        end
        If specified write out to disk alternative vector-based formats (KML,SHP,GPX) for each
        version v of object i
    end
end
```

Algorithm 1: A simple algorithm to download and store history of objects from OSM

All versions v0 to vi are stored in a PostGIS spatial database. The elements of tuple Pi above are stored using primitive data types. Every version v of a given object is stored as a geographic object in PostGIS. The PHP code creates a Well-Known-Text (WKT) representation of the object v and inserts this into the database. The OSM-ID and current version form a composite primary key for each version of each object in the database. A key bottleneck of this approach is the download of the spatial data for each version of the history of each object. This requires repeated HTTP GET calls to the OSM API. Downloaded nodes are cached in a temporary database table to assist in this processing. However, despite this bottleneck this approach is more flexible and efficient than the alternative download of raw OXM for a large region and subsequent insertion into a spatial database.

3. CONCLUSIONS

Our software removes the "grunt work" from transforming OSM-XML to a spatial database model. This allows GIS researchers to focus their research resources on analysing and working with the OSM-XML. Currently the software is command line driven. As part of our future work we are investigating the development of a web-based tool for animation of editing histories in OSM for small groups of objects (ie housing estate). When the software has been adequately tested and reached an acceptable level of maturity it will be shared openly with the OpenStreetMap community and the GIS research community.

ACKNOWLEDGEMENTS

Dr. Peter Mooney is a research fellow at the Department of Computer Science NUI Maynooth and he is funded by the Irish Environmental Protection Agency STRIVE programme (grant 2008-FS-DM-14-S4). Dr. Padraig Corcoran is a lecturer and post-doctoral researcher also at the Department of Computer Science NUI Maynooth. Dr. Corcoran is part of STRAT-AG which is a Strategic Research Cluster grant (07/SRC/I1168) funded by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

REFERENCES

Comber, A., Fisher, P. and Wadsworth, R. (2005). What is land cover?, Environment and Planning B: Planning and Design 32(1): 199–209.

Corcoran, P., Mooney, P. and Winstanley, A. C. (2011). Planar and non-planar topologically consistent vector map simplification, International Journal of Geographical Information Science x(x): To Appear.

GeoFabrik (2010). Geofabrik: Download server for openstreetmap data, Web Based Download Application: http://download.geofabrik.de/ - last checked December 2010.

Haklay, M., Basiouka, S., Antoniou, V. and Ather, A. (2010). How many volunteers does it take to map an area well? the validity of linus' law to volunteered geographic information, The Cartographic Journal 47(4): 315 – 322.

Harrie, L. and Stigmar, H. (2010). An evaluation of measures for quantifying map information, ISPRS Journal of Photogrammetry and Remote Sensing 65(3): 266 – 274.

Heipke, C. (2010). Crowdsourcing geospatial data, ISPRS Journal of Photogrammetry and Remote Sensing 65(6): 550–557.

Mooney, P., Corcoran, P. and Winstanley, A. (2010a). A study of data representation of natural features in openstreetmap, in R. Purves and R. Weibel (eds), Proceedings of GIScience 2010: The 6th International Conference on Geographic Information Science, Published online at http: //www.giscience2010.org, p. p150.

Mooney, P., Corcoran, P. and Winstanley, A. C. (2010b). Towards quality metrics for openstreetmap, Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, ACM, New York, NY, USA, pp. 514–517.

Over, M., Schilling, A., Neubauer, S. and Zipf, A. (2010). Generating web-based 3d city models from openstreetmap: The current situation in germany, Computers, Environment and Urban Systems 34(6): 496 – 507. GeoVisualization and the Digital City - Special issue of the International Cartographic Association Commission on GeoVisualization.

Triglav, J., Petrovic, D. and Stopar, B. (2011). Spatio-temporal evaluation matrices for geospatial data, International Journal of Applied Earth Observation and Geoinformation 13(1): 100–109.