Micro tasking, an efficient method for data acquisition for digital maps?

Anne Sofie Strand Erichsen

November 11, 2016

Contents

List of Figures

Abstract

TB!

1 Introduction

Voluntary acquisition of data for digital mapping has become quite popular over the last years. This is also a cornerstone for Open Street Map (OSM), which is a global open access database for digital maps. To establish data sets (for maps) is a comprehensive task. By dividing the work into many small sub-tasks, each of these tasks may be possible to handle by voluntary enthusiasts. This is called Microtasking.

The Norwegian government has indicated that the most detailed map data for Norway (FKB) may become freely available within a few years. If so, these data may be included in the database of OSM.

During this paper the student is suppose to:
- Study relevant literature
- Study how Microtasking is working
- Explore how data from FKB can be mapped into OSM
- Investigate if Microtasking is a possible method for including FKB in OSM

2 | Characteristics of Open Street Map

2.1 Introduction

The OpenStreetMap is one of the most impressive projects of Volunteered Geographic Information on the Internet[Neis and Zipf, 2012]. Until recent the mapping of the Earth was preserved highly skilled, well-equipped and organized individuals and groups. One important happening was in 2000 when Bill Clinton removed the selective availability of the GPS signal **??**. This change improved the accuracy of simpler, cheaper GPS receivers so that also ordinary people could start mapping their movements. OpenStreetMap was founded in 2004 at University College London by Steve Coast. The goal was to create a free database with geoxgraphic information of the world [Neis and Zipf, 2012]. Back in 2004 the geographic data was expencieve and hard to get access to.

The OSM project stands out from other data sources mainly because its free to use and its released under a license that allows for pretty much whatever the user wants to as long as the user mention the original creator and the licence[Chilton et al.,]. The most common contribution approach is to record data using a GPS receiver and edit the data using one of the free and available OSM editors [Neis and Zipf, 2012].

Today the world has a need for instant information, particularly in crisis situations [Chilton et al.,]. Here OpenStreetMap is the leading global example of the effictiveness of crowdsourcing of geodata. The project are changing the way individuals and organisations are thinking about the collection process, purchase and use of geodata [Chilton et al.,]. Crowd sourced geographic data has characteristics or advantages of large data volume, high currency, large quantity of information and low cost [Wang et al., 2013].

2.2 Culture

OSMhas no notability rule, an arbitrary amount of detail is possible, but somebody has to maintain it!

2.3 Structure

OpenStreetMap uses a topological data structure. This sturcture includes three basic components nodes, ways and relations. Nodes are points with a geographic position stored as coordinates (Lat, long) according to WGS84. Ways are lists of 2 or more nodes, representing a poly-line or polygon, used to represent streets, rivers, among others [Debruyne et al., 2015]. A relation is a multi-purpose data structure that documents a relation between two or more components. To add metadata to geographic objects OpenStreetMap uses Tags. Tags consist of two items, a key and a value of the form key=value. The key is used to describe the topic, category or type of feature, while the value describes the details of the specific form of the key specified. A example of a key-value pair can be building=church, here the key is building and the value is church, this is a building that was built as a church.

The norm in OSM is to try to map new data with existing tags. Good practice is to search for tags, or Map Features, on different OSM wiki-sites. On the tags you like wikipage they recommend different sites, but points out taginfo as the most useful site. Taginfo is a website created for finding and aggregating information about OSM tags, it covers the whole planet and is updated daily. The web page list tags used in the database and also inform on how often they have been used. Also, Taginfo lists other tags which have been used in combination with the tags you searched for. Some countries also have their own taginfo web pages, like Ireland, Great-Britain and France, Norway do not have their own taginfo web page.

Verifiability: From a given scenario, a tag/value combination is verifiable if and only if independent users when observing the same feature would make the same observation every time..

2.4 Organizational

Organization and communication

The OpenStreetMap Project is supported by the OpenStreeMap Foundation (OSMF) which is a UK-registered non-profit organization. The foundation was founded in 2006 and consists of members from all over the world, as of December 2015 consist of 350 normal-, 351 associate- and 18 corporate members [OSMF, 2015]. OSMF include a board of seven members and is critical to the ongoing function and growth of the OpenStreetMap project [OSMF,]. The foundation has the responsibility for the servers and services necessary for hosting the OSM project. Also, they support and communicates with the working groups, and delegates tasks that has to be done, like Web site development etc.

A person can contribute to the OSM project without being a member of the foundation. The project has over 3 million registered users [OpenStreetMap, 2016] who are collecting and updating data. The crowdsourced data are then released under the Open Database License, *"a license agreement intended to allow users to freely share, modify, and use this Database while maintaining this same freedom for others"* [ODbL,]. Users can edit maps through different tools made by different OSM contributors. One tools is called iD and is the default web browser editor written by MapBox. There are also desktop editing applications like JOSM and Merkaartor which are more powerful and better suited for advanced users.

Communication is country-mailing lists, wiki-pages and conferences. State of the map is the main OSM conference. Number of mappers and organizations are constantly increasing, what started as a crazy hacker project is now a vital part of the global data ecosystem. The community are constantly developing new ways to contribute. Users can join a mapathon in their hometown, they can sit at home adding data,

The degree to how you can get involved in OSM is so deep, mapping, software processes etc. - keeps people interested. One problem is the communication through the different groups, the energy level is not high enough. Lots of people exited about communication, everyone have a obligation to show the users their different possibilities.

Communication through mailing lists works for the people who subscribes to that list, but with over 150 different list it is impossible for an interested user to stay updated with all the latest achievements. What is happening in the mailing lists has to be available to everyone in the community. A solution to this is called

weeklyOSM. It started in 2010 in German, who alone has 50 different mailing lists. The weeklyOSM team are scanning mailing lists, twitter, blogs and so on. There are a international team translating the German blog into different languages. Little by little the rest of the community gets involved, translating the blog into languages thats not supported.

There are different groups creating the different versions. The goal is to integrate more languages. There are a lot of work every week, hard to find volunteers.

2.5 File format, .osm files

The .osm file format is specific to OpenStreetMap and it is not easy to open these files using GIS-software like QGIS. The file format is designed to be easily sent and received across the internet in a standard format. Therefore .osm files are easily obtained, but using the files directly to do analysing and map design is not easy. The .osm files are coded in the XML format. It is recommended to convert the data into other formats when using the files source.

The file is very difficult

3 | FKB building, norwegian governmental data

This chapter will give an introduction on the SOSI standard and the SOSI file format. This is a standard developed by the norwegian mapping authority and is the largest national standard for geographic information. SOSI is widely used file format for Norwegian mapping [Kartverket, a]. FKB is a collection of datasets containing detailed vector data for Norway and comes in SOSI format. Since this paper looks at FKB building dataset this chapter will describe the most common and valuable features for OpenStreetMap within this dataset. Some features are not relevant for import into OpenStreetMap. The chapter ends of with an evaluation of SOSI, looking at its advantages against OpenStreetMap.

3.1 SOSI

The national standard for geodata in Norway is called SOSI, created by the norwegian mapping authority. Geodata is map data stored in a digital format so that we can produce maps from it. The standard are based on international standards, primarily the NS-EN ISO19100-family of standards [Skogseth and Norberg, 2014]. The SOSI standard is implemented in the SOSI format, a norwegian geospatial vector data format. The SOSI data consists of point-, line (*kurve*) - and area (*flate*) features. Point feature is only one single vertex, given in north and east coordinates with or withouth height. Line feature is two or more vertices, but the first and last vertex is not equal. A area feature is three or more vertices, where the first and last vertex are equal.

The norwegian map authority has established a general feature directory *(objekt*

katalog) in connection with the SOSI standard. The purpose of a feature directory is to specify feature types and associated properties that is general within a dicipline or across multiple disciplines [Kartverket, 2016b]. The directory covers around 50 disciplines (2014) [Skogseth and Norberg, 2014]. SOSI version 4 changed method, from modelling point, line and area to modelling feature types in the real world (buildings, roads, boundaries etc.) [Skogseth and Norberg, 2014]. For instance, a road will have many associated properties in addition it can be located as a line, this is how SOSI version 4 models its data*.

SOSI data can be presented on four different levels, each level represents a different data quality. A SOSI dataset contains information on multiple levels. First comes the *Head* containing shared information about the dataset, this information applies to all the data. Then comes the *Data itself* containing properties and location coordinates (N, E and height). The SOSI dataset is finished with an *End* which is the end of the data series [Skogseth and Norberg, 2014].

3.2 FKB and data quality

FKB, *felles kartdatabase* in norwegian, is a collection of structured datasets that contains the most detailed vector data of Norway. FKB data is collected through a collaboration called Geovekst. This is a collaboration between the nowergian mapping authority, the norwegian road authority, Telenor, Energy Norway, the Norwegian Association of Local and Regional Authorities, ministry of Argiculture and the Norwegian Water Resources and Energy Directorate. FKB data comes as vector data in SOSI format [Kartverket, 2011].

The FKB standard describes which features that is included in the mapping and the accuracy of the objects. There are specified four FKB standards, FKB-A, FKB-B, FKB-C and FKB-D [Kartverket, 2011]. FKB-A is the most detailed, containing good three dimensional data description and has high standards on accuracy (5-20 cm) and content. Most common used in city centers. FKB-B is also detailed with an accuracy of 20 - 30 cm. Mostly used in urban areas. FKB-C is used for overview planning and management (forvaltning) with an accuracy of 0.50 - 2.00 m. FKB-D are areas not covered in the three other standards, like mountain areas, and has a broad accuracy of 5 - 100 m. Today, maps should always be produced after the FKB standards [Skogseth and Norberg, 2014].

This paper will look into the FBK building dataset. The data consist of both point,

line and area and contains 24 different feature types [Kartverket, 2013a]. The data is established and kept up to data by using photogrammetry. In some cases the data are established by using land surveying [Kartverket, 2013b]. Building points are tranferred from the Cadastre (*Matrikkelen*). The data are delivered in the official reference system for each municipality.

Today FKB data is saved piecewise within each municipality. The data is collected in a database at the norwegian mapping authority which is updated one or two times a year. This is not the optimal solution. A goal is to gather all FKB data, from every municipality, into one central database where all updates will be made directly to this central database. The goal is to have 80% of norwegian municipalities connected to the database within 2018 [Kartverket, 2016a].

Possibilities for 3D representation of buildings from the FKB standard varies. Some feature types in the FKB dataset have a level of detail attribute called TRE D NIVÃĚ where they usually have six levels. Level 0 is only 2D, limited to the ground floor. Level 1 is buildings represented as blocks with a flat roof. Height of the roof is either the minimum, maximum or average of ceiling height around the building. Recognizability are not great, especially for apartment buildings. In level 2 the main shape of the roof are maintained with use of ridge lines and break lines. Photogrammetric data capture for FKB-A, -B and -C standards provides buildings with level of detail similar level 2. Level 3 includes added features as dormers, balconies, larger chimneys etc. Gives a better visual quality and more appropriate basis for analyzes. Photogrammetric data capture for FKB-A and -B standards provides level of detail similar level 3, but details are different for the two standards. Level 4 is a high quality model of buildings, not supported in the SOSI standard building model [Kartverket, b]. Level 5 is a high quality mode of a building both outside and inside, not supported in the SOSI standard building model [Kartverket, b]. FKB-A and FKB-B features describing the main roof has to be at least level 2 of detail and features describing details located at the roof with at least level 3 of detail.

3.3 Features - FKB building

There are 24 different feature types in the FKB building dataset. Where two features are point data, three features are area (*flate*) data and the rest is line (*kurve*) data. The features are grouped into four categories, building and building refinement

(*bygningsavgrensning*), descriptive building lines, building appendage (*bygningsvedheng*) and lastly roof covering (*takoverbygg*).

There are features that are not as relevant for OpenStreetMap. Adding all buildings as point data does not seem relevant. The only point feature types are building and assistandpoint3D (*hjelpepunkt3D*). The building feature comes as both point and area, containing exactly the same attributes. Adding building as a point therefore seems unrelevant, it will not add additional information to OSM. Searching for the hjelpepunkt3D feature in a arbitrary FKB dataset in one municipality returns only 18 rows with 1499 rows in total, therefore this feature will not be helpfull for 3D modelling buldings, and also not relevant when excluding building point feature. Therefore no point data will be relevant for import.

Looking at FKB building dataset over Trondheim municipality gives some indication of the most common feature types and help determining which features should be prioritized in the convertion between FKB building SOSI file and OSM file format. Lets start with line data, which in Trondheim consist of 618 710 rows. In Trondheim there are 158 917 roof edge (*takkant*) features and is the most common line feature in this municipality. This feature is the building's exterior roof surface refinement (*avgrensning*). See figure 3.1 and 3.2 for examples on how to use this feature. The second most common line feature is ridge line (*Monelinje*). There are 103 488 rows with this feature in Trondheim. Ridge line is the line describing the horizontal bending line / breakline (*knekklinje*) on top of the roof, the highest peak at the roof. See figure 3.1 and 3.2 for example of how it is used. A minimum goal for the FKB mapping team is to map ridge lines on every building and can explain the hight frequency of this feature [Kartverket, 2013b].

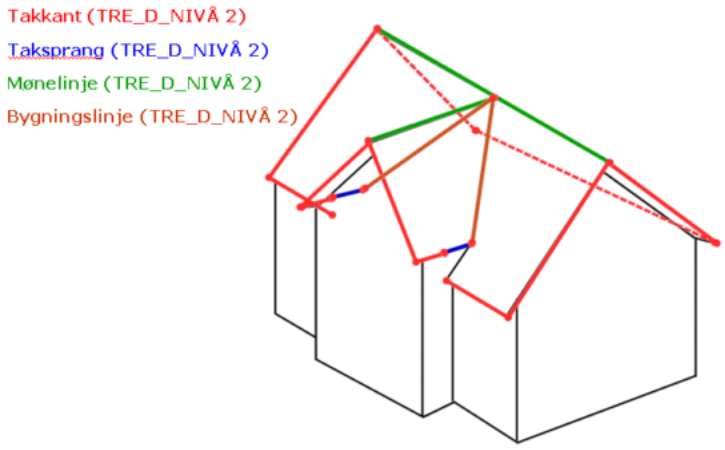

Figure 3.1: Roof edge (red), roof overhang (blue), ridge line (green) and building line (brown) [Kartverket, 2013b]

Third most common line feature in Trondheim is roof overhang (*taksprang*) with 96 436 rows. This feature describes the top of the roof edge inside the building shell, not on the outside edge which is the roof edge feature. For an example see figure 3.1 and 3.2. This feature should be mapped where height difference between two roof levels is larger than the tolerance of the FKB-data. The feature are in the descriptive building lines category. The line always follows the roof edge.

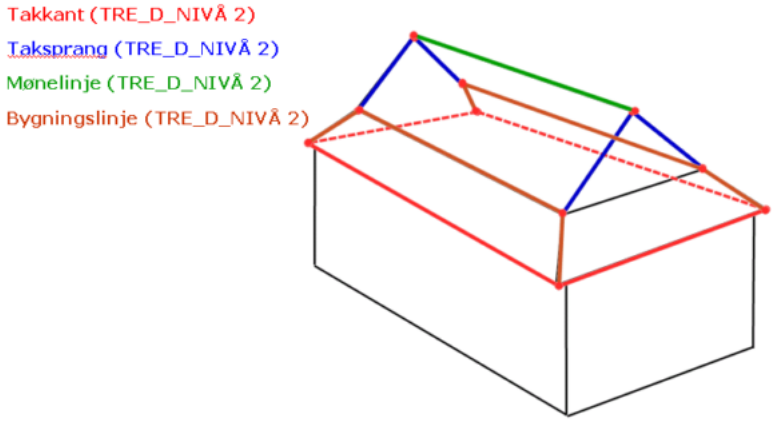

Figure 3.2: Roof edge (red), roof overhang (blue), ridge line (green) and building line (brown) [Kartverket, 2013b]

Fourth most common line feature in Trondheim is roof overhang bottom (*TakSprangBunn*) with 91 281 rows. This feature describes lines located at the bottom of a roof edge within a building mass (*bygningskropp*). The feature are under the descriptive building lines category. In figure 3.3 the blue line shows where a roof overhang bottom line can be drawn. As shown in figure 3.3 roof overhang bottom lines should, if possible, have equal coordinate values (N, E) as the corresponding roof overhang. This is visualized with the dashed black lines on figure 3.3. In figure3.3 the red and blue lines are from the original figure in [Kartverket, 2013b], the green line is added afterwards to visualize roof overhang in the same figure.

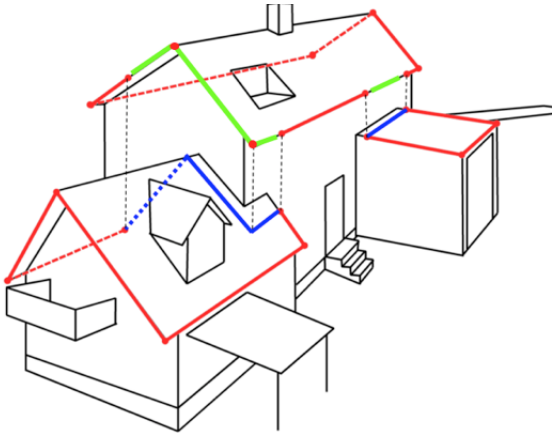

Figure 3.3: Roof overhang bottom (blue), roof edge (red) and roof overhang (green) [Kartverket, 2013b] + me

Fifth most common line feature in Trondheim is veranda and includes veranda, terrace, balcony and loading ramp [Kartverket, c]. If the area mapped is following FKB-A standard, veranda features down to two square meters are added. If the area mapped is following FKB-B standard, veranda features down to six square meters are added. Veranda features has a attribute value MEDIUM that described if it is located on the roof (MEDIUM = B), on the outer wall (MEDIUM = L) or on terrain (MED IUM = T). This attribute is helpful when making 3D models of the buildings. Height attributes can either be reference at the top of railing (used for medium B) or at floor level (used for medium T). When the feature has attribute medium L its optional which height reference to use. See figure 3.4 for example of veranda features. Veranda features are under the building appendage category.

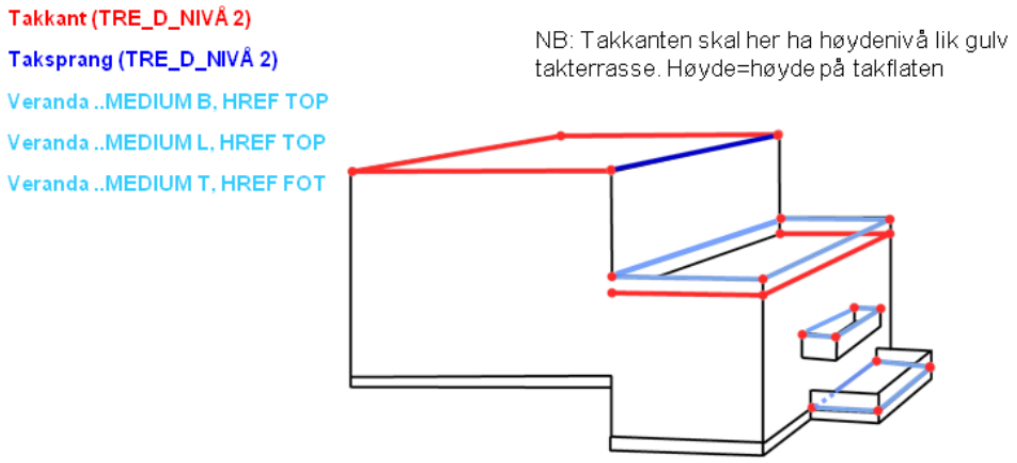

Figure 3.4: Roof edge (red), roof overhang (blue) and veranda (turquoise) [Kartverket, 2013b]

Sixth most common line feature in Trondheim is building line (*bygningslinje*) with 53 255 rows. The feature is used when describing building details which are within a roof perimeter, and which cannot be described by other feature types. If the area is covered with FKB-A standard the building lines should be drawn on objects that are two square meters and bigger,

It is impossible, at least in FKB, to create a specification of registration of buildings that are completely accurate. The buildings will always be subject to some generalization. This can be seen in the figures used in this section.

3.4 Fordeler med SOSI mot OSM

Ikke har et poligon direkte, flate er bare referanse til linjene

Point, line, area (flate)

4 | OSM import methods

The traditional way to contribute data to the OpenStreetMap project is through active users who use their GPS to track roads and their local knowledge to add information about geographic regions to the OSM database [Zielstra et al., 2013]. Users also digitalize aerial photos. Cheaper GPS receivers and more available satellite imagery with better resolution makes it easier for users to contribute [Chilton et al.,]. The number of active users in different regions varies a lot, making some regions on the OSM map full of data while others are almost empty. This led to a second approach for getting data in to the OpenStreetMap database, bulk imports [Zielstra et al., 2013]. Bulk import is the process of uploading external data and were meant for initial/preliminary object class uploads, so only if the object were none existing in the area [Zielstra et al., 2013]. Its a good alternative for countries or regions with few or none active users. Through the years different import methods has been developed. This chapter will evaluate the most common methods.

4.1 Fully automatic import script

Creating a script that automatically imports big datasets into OpenStreetMap, a bulk import, is not encouraged in the OSM community [Zielstra et al., 2013]. This becomes clear when reading the wikipages about import. A bulk import is suppose to be a supplement to user generated data. The user generated data and the users ability to work is always the priority [OpenStreetMap, c]. A fully automatic import do automated edits to the OpenStreetMap database with little, if any, verification from a human. Automatic edits is changes that has no or very limited human oversight [edits OpenStreetMap,]. This kind of edits must follow the Automated Edits code of conduct [OpenStreetMap, a]. The policy was created to prevent damaging acts on the database and ignoring it will result in the import be treated as vandalism.

An example of a bulk import was the TIGER import. The Topologically Integrated Geographical Encoding and Reference system (TIGER) data was produced by the US Census Bureau and is a public domain data source. The bulk import was completed in early 2008 [Zielstra et al., 2013], populating the nearly empty map of the United States. The TIGER dataset was not perfect and had it's faults, but it was better than no data at all [Willis, 2008]. The import mainly focused on data containing the general road network for the US [Zielstra et al., 2013].

Reading through the OpenStreetMap mailing thread called talk, it becomes clear that this import is an automatic edit with very limited oversight. *"Please don't upload Northampton County, [..] I've mapped my entire town there [..]"* [Mielczarek, 2007]. Active OSM mappers on the mailing thread tries to save their manuall work . Users not attenting the mailing list had limited, if any, possibilities of their work beeing saved through the import. The mappers importing TIGER data tried not to override existing data. They started on empty states but reading though the "TIGER, which states next?" mailing thread the decisions on which state to be imported next was only based on the people attending the conversations. *"I believe I'm the only one out here in Nebraska [..] Feel free to override my edits"* [Bishop, 2007]. The import process did not have any requirements on what they should do with existing data. OSM mappers added the TIGER data county by county, state by state. The OSM user Dave Hansen, one of the most active users during this import, created a text file with his upload queue [Hansen, 2007b] where he added states and counties after requests from users [Hansen, 2007c][Hansen, 2007a]. The import team did not have any validation or correction methods or routines. In the aftermath of the import tagging errors have been discovered. For instance, the TIGER data groups residential, local neighbourhood roads, rural roads and city streets into one road class, while OSM uses a more refined data schema with different highway tags for different road classes [Zielstra et al., 2013].

On the TIGER OpenStreetMap wikipage, last updated August 2016, they say it is unlikely that the TIGER data ever will be imported again. The main reason is the growing US mapping community, their mapping is often better than the TIGER data. *"Do not worry about getting your work overwritten by new TIGER data. Go map!"* [OpenStreetMap, 2007]. A new bulk import with updated TIGER data can overwrite existing, more precise data. The TIGER data are of variable quality, poor road alignment is a huge problem and also wrong highway classification. Many hours of volunteer work could be lost and this is something the community want to avoid. The bulk import in 2007 got the United States on the OSM-Map and saved the mapping community a lot of time finding road names etc. *"TIGER is a skeleton on*

which we can build some much better maps" [Willis, 2007]. On the negative side the project kept the US mappers away, they were told for years that their work was no longer needed after the TIGER upload was complete. But the presence of TIGER data ended up requiring a lot of volunteer help, they needed help fixing errors like the poor road alignments and wrong tagging.

Bulk imports are overall not recommended today, but have been helpful as well. In the Netherlands bulk imports have met little resistance, mainly because the imports are done by dedicated OpenStreetMap mappers who knew the OSM import guidelines. Arguments in favor of bulk imports say that a map that already contains some information is easier to work on and can help lower the entry barriers for new contributors. Another argument is that a almost complete map is more attractive for potential users, that again can encourage more use of OSM data in professional terms [Exel van, 2010]. But a huge minus to bulk imports are the data aging, since the data being imported often already is a few years old and updating it takes time, often years. The TIGER import was data from 2005, but the import finished in 2008 [Zielstra et al., 2013]. Between the time of first import and update the community have fixed bugs, added important metadata, the community would not want to loose that data/information.

Today there are huge amounts of object types in OpenStreetMap. Bulk imports with limited human interaction do often end up overriding existing data, which is one of the "don't do" points on the import guidelines list on the osm wiki page. OpenStreetMap do not have layers, so data on top of data makes it very difficult to organize and find the data.

4.2 Guided automatic import

Fully automatic import of huge amounts of data is discouraged in the OSM community, so another approach is guided automatic import. The OSM community encourages people to import only small amounts of data at a time and only after validation and correcting errors [Mehus, 2014]. This method was used when the OSM-community in Norway got approval from the Norwegian map authority to import N50 data [Kihle, 2014]. N50 is the official topographic map of Norway. The import process is described on the OSM wikipage. The mapping team would import one municipality at a time. Each municipality dataset were divided up in a .15 deg times .15 deg grid changesets, each changeset contained from five thousand to twenty thousand

elements [OpenStreetMap, 2014]. The N50 import was an community import, but only experienced user were encouraged to import the data [Mehus, 2014].

The N50 release was good news for Norway, since regions, especially in northern parts of Norway, had very little data because of few active OSM mappers. This import would then increase the quality of OpenStreetMap in much bigger parts of Norway kilde. Its a huge dataset so they had to import it with caution, not everything in the dataset were relevant for OpenStreetMap, this is noted by the OSM user Solhagen hrefhttp://wiki.openstreetmap.org/wiki/User:Solhagenkilde.

The data was preprocessed by the OSM user tibnor and uploaded to a google drive folder available for everyone. He started on the preprocessing early 2015. Late 2013 the OSM user gnonthgol* created sosi2osm script for convertion between the norwegian dataformat SOSI and OSM file format kilde. This script is the recommended way of converting the N50 sosi files to the OSM file format. In may 2015 tibnor released a JOSM plugin for the N50 import of rivers/streams to make it faster to check and fix directions. The import process was managed through a wiki page. Here users wrote their name and progression, startdate and enddate of the imported municipality. Elements which needed manual inspection or validation was tagged with "Fix-me" and a description on what to do. They used a python script ("replaceWithOsm.py") to merge N50 data with existing OSM data, adding source=Kartverket N50 tags on the new data. Elements that already exists in OSM, that conflicts with the new data, is marked with FIXME=Merge. Here the user has to search for the conflicting elements and correct the errors manually. Then the data can be uploaded to OSM.

The N50 import was initially stopped in May 2014 by Paul Norman. The norwegian OSM group started importing the N50 data before consulting with the OSM imports mailing list, which is required. They were also importing the data without the proper approach. This was pointed out by DWG member Paul Norman [Mehus, 2014]. DWG is the data working group, created in 2012, and they are authorized by the OSMF to detect and stop imports that are against the import guidelines [OpenStreetMap, b]. The Data working group reverted the import because of technical problems and errors in the import [Hagen, 2014]. This was a step back for the Norwegian OSM team, they had to start over again. The DWG stopping the import in 2014 was probably for the best, the DWG group has much experience with automated imports.

The N50 import has been time consuming. It started in 2015 and is still not finished, even though most of the municipalities are imported. The import process was carried out according to the import guidelines. Without the DWG group the import would

probably end up as a automated edit with no proper validation process. This is one example of importing existing data into OpenStreetMap is a time consuming job. There are guidelines to follow, a lot of validations to be done.

4.3 Import based on microtasking (LA-buildings methodolo)

The N50 import was a good start towards microtasking an import of huge amounts of data. They divided every municipality into .15 degree .15 degree grid changesets and imported one changset at a time. Both the New York bulding import, finished in 2014, and Los Angeles building import, not finished, took this mindset to the next level, creating a Tasking manager interface specific to this import, among other initiatives. The LA-building team created a custom tasking manager to coordinate the LA County building import [OSM Tasking Manager,], while the NY-team used the original OSM Tasking Manager.

The OSM Tasking Manager was created in the aftermath of the Haiti earthquake. This innovation coincided with the growing popularity of microtasking as a solution to manage distributed work [Palen et al., 2015]. The Tasking Manager tool was initially created by the newly formed HOT (Humanitarian OpenStreetMap) to help mappers more efficiently coordinate simultanrous work [Palen et al., 2015]. The purpose of the tool is to split the mapping into smaller parts, the parts are mapped independently and should be completed in a short timeperiod [HOT,]. The tool is open source and its code is available on Github, making it easier for other projects to create their versions of it.

The Guided automatic import from 4.2 we saw that dividing datasets into smaller parts makes the import easier to, among others, distribute the workload between experienced users. OSM Tasking manager takes this approach further by among others, offering a graphical user interface around the import. The tasking manager contains important information. Like a description about the import, instructions on how to do the import and important tags etc. It provides an easy way of downloading a dataset, bounded by a grid, into JOSM or id editor. The interface contains a map visualizing which grids are done, which are occupied and how far they have come in the import process. The tasking manager use colors to inform the user if the grid is done, locks it if someone is already working in the grid and also the user easily can see the commit story when they click on a grid. This is important information to the mappers. They can easier work at the same time without worrying about

overlapping import.

The New York building import took 10 months, finishing in June 2014. The project started as a community import, but underestimating the import complexity and time spent training and supporting new mappers they restarted a few months in, loosely forming a group around the project [Barth, 2014b]. The group consisted of volunteer mappers and employees from the Mapbox team. This grouping made coordination easier and also made it easier to ensure proper training [Barth, 2014a]. More than 20 people spent more than 1500 hours, importing 1 million buildings and over 900 000 addresses [Barth, 2014b]. Common issues during import was written on the Github page. The New York city data was first converted into OSM format, then cut into byte sized blocks which was reviewed and imported manually through the tasking manager, piece by piece. An important validation step was that a different person than the original importer validated the data, reviewing it for errors and cleaning up when needed [Barth, 2014b]. The NY-team developed tasking manager 2 so that tasks can be shaped as arbitrary polygons, rather than automatically squares.

The LA building import started in 2014. Two OSM enthusiasts started on the project, Jon Schleuss and Omar Ureta. They used code from the NY building import and adopted it to their needs. After a while Mapbox joined in on the import, a important step for the project. Mapbox helped with programming important scripts, converting the data to osm files, creating block groups of the data. First challenge was to decide which datasets to import. They ended up neglecting address data, which would *delay the project with 1 year or 2* - quote Jon Schleuss, adding only building outlines and building info (assessor data) [Schleuss et al., 2016]. They merged and cleaned the datasets, splitting them into blocks and serving the data to the tasking manager. They used mapathons to get the import started. MaptimeLA was the organizer and they also created tutorials for new mappers who joined the team. The first mapathons started with JOSM training to new mappers. Evolving to only arranging import mapathons, or import parties. Through mapathons is was easier for inexperienced mappers to contribute, here they could get the necessary training. When importing data mappers always have to examine for possible conflicts between existing and new data. If a conflict is found, and the mapper doesn't know how to deal with it, they can flag the .osm file and a more advanced user will look at it. The task will then be finished by someone else.

A big difference between the NY building import and LA building import was that the NY team ended up only allowing some OSM users to import. The NY-import was planned as a community import, but underestimating the import complexity and time spent training and supporting new mappers they restarted a few months

in, they loosely formed a group around the project [Barth, 2014b]. The LA building team allowed everyone to join the import. To keep an track of mappers the LA team created a list where the volunteers had to write their import username [**?**]. Doing the import job during mapathons is a good idea, it makes it easier to have an overall control over the import process.

In NY building import, when a error was discovered that required updates to already imported data they had to do an automated edit. Updating existing data manually was very time consuming. Updating OpenStreetMap data programatically, with a script, is according to Alex Barth in Mapbox, crucial for a successful import. The LA community have pointed out errors that need to be fixed, for instance is Garage incorrectly labelled as houses and condos have been tagged as house not apartments. Reading through the different issues reported on LA building github page are some errors found in already imported data fixed manually and others through scripts. They created a Maproulette challenge on at least one issue, fixing split buildings. The LA team created a script to detect all split buildings and then each detected building were available as a task in Maproulette [Sambale, 2016]. Each task was imported into JOSM where they used the plugin *Auto-tools* to merged the split buildings. Maproulette is a gamified approach for fixing OSM errors breaking the error into micro tasks [OpenStreetMap, d].

When using the OSM tasking manager the dataset has to be divided into smaller parts, so that each part has a manageable size for manual uploads. Each part represents one task, and it is important that each task are small enough so that it can be completed in a reasonable time [**?**] **. The NY team created a python script (chunk.py) to divide the data into smaller parts. The script divided the data into the New York City voting districts, there are in total 5258 voting districts, creating 5258 tasks in OSM tasking manager. This was an arbitrary choice, determined by the import team [Barth, 2014b]. The LA building mapper Alan McConchie opened a issue on labuildings github page asking *How to divide up the tasks?* [McConchie, 2014]. He suggested using census block groups in each county, the grouping gave suitable sized areas for the tasks. Census blocks alone would be too granular tasks, and next level there are tracks, which would result in too big tasks. They used the same script as NY building import (chunk.py).

The OSM community needs a script which can divide any dataset into smaller blocks, independent on the location of the data. There shouldn't be necessary to spent time on finding existing dataset which can be used to divide the import data into tasks.

4.4 Evaluation

OpenStreetMap is a large community dependent on active users adding data in their geographic regions. The users have different perspectives on importing data through scripts. In empty regions with few active users a bulk import can be good way of developing the OSM project. But in regions with large amounts of data, there is no agreement on whats best*. Little research has been done on countries where the OSM project relied on data imports to fill the map. The OSM community is undecided on the benefits of bulk imports for the OSM project, especially for areas such as the US where large gorvermental data are freely available [Zielstra et al., 2013].

Validation of the data being imported takes time. Another dimension to bulk imports, with validation especially through tasking manager, is that unrelated issues in the same area gets fixed by the mappers. During the New York building import they fixed 5,000 unrelated map issued along the way [Barth, 2014b].

Do the OSM Tasking Manager tool make the import of data into OSM easier for the users? Of course it makes the coordination easier. But looking in the technical perspective maybe not. Or it just tells us that importing data to OSM following all the import guidelines will never be easy. The users that do the import need technical background and deep understanding on how OSM works.

During the LA import the LA county released new data. This new data was used for rest of the import, but they did not update the already imported regions. How old was the NY building data?

Bibliography

[Barth, 2014a] Barth, A. (2014a). [Imports-us] Restarting NYC building and address imports.

[Barth, 2014b] Barth, A. (2014b). OpenStreetMap | lxbarth sin dagbok | Importing 1 million New York City buildings and addresses.

[Bishop, 2007] Bishop, D. (2007). [OSM-talk] TIGER, which states next?

[Chilton et al.,] Chilton, S., Building, F., and Burroughs, T. CROWDSOURCING IS RADICALLY CHANGING THE GEODATA LANDSCAPE : CASE STUDY OF OPENSTREETMAP.

[Debruyne et al., 2015] Debruyne, C., Panetto, H., Meersman, R., Dillon, T., Weichhart, G., An, Y., and Ardagna Afostino, C. (2015). On the Move to Meaningful Internet Systems: OTM 2015 Conferences. *Springer International Publishing*.

[edits OpenStreetMap,] edits OpenStreetMap, A. Automated edits - OpenStreetMap Wiki.

[Exel van, 2010] Exel van, M. (2010). Data Imports In OpenStreetMap âĂŞ Love âĂŸEm Or Loathe âĂŸEm? | oegeo.

[Hagen, 2014] Hagen, S. O. (2014). Re: [NUUG kart] Møsvatn.

[Hansen, 2007a] Hansen, D. (2007a). [OSM-talk] TIGER update (need more suggestions).

[Hansen, 2007b] Hansen, D. (2007b). [OSM-talk] TIGER upload queue (maps of entire US).

[Hansen, 2007c] Hansen, D. (2007c). [OSM-talk] TIGER upload queue (maps of entire US).

[HOT,] HOT. HOT Tasking Manager - About.

[Kartverket, a] Kartverket. SOSI | Kartverket.

[Kartverket, 2011] Kartverket (2011). SOSI Del 3 Produktspesifikasjon for FKB âĂŞ Generell del Side 2 av 55.

[Kartverket, 2013a] Kartverket (2013a). Samletabell over objekttyper i FKB.

[Kartverket, 2013b] Kartverket (2013b). SOSI Del 3 Produktspesifikasjon for FKB âĂŞ Bygning.

[Kartverket, 2016a] Kartverket (2016a). Innføring i kommunene | Kartverket.

[Kartverket, 2016b] Kartverket (2016b). SOSI-standard del 2 Generell objektkatalog.

[Kartverket, b] Kartverket, S.-s. TreDNivå - Geonorge objektregister.

[Kartverket, c] Kartverket, S.-s. Veranda - Geonorge objektregister.

[Kihle, 2014] Kihle, K. (2014). [NUUG kart] Bruk av datasett fra Kartverket i OpenStreetMap.

[McConchie, 2014] McConchie, A. (2014). How to divide up the tasks? #4 Github issues, LA buildings.

[Mehus, 2014] Mehus, T. (2014). [Imports] N50 imports from Kartverket (The Norwegian Mapping Authority).

[Mielczarek, 2007] Mielczarek, T. (2007). [OSM-talk] [OSM-dev] TIGER, which states next?

[Neis and Zipf, 2012] Neis, P. and Zipf, A. (2012). Analyzing the Contributor Activity of a Volunteered Geographic Information Project âĂŤ The Case of OpenStreetMap. *ISPRS International Journal of Geo-Information*, 1(3):146–165.

[ODbL,] ODbL. Open Database License (ODbL) v1.0 | Open Data Commons.

[OpenStreetMap, a] OpenStreetMap, C. Automated Edits code of conduct - OpenStreetMap Wiki.

[OpenStreetMap, b] OpenStreetMap, D. Data working group - OpenStreetMap Wiki.

[OpenStreetMap, c] OpenStreetMap, I. Import - OpenStreetMap Wiki.

[OpenStreetMap, d] OpenStreetMap, M. MapRoulette - OpenStreetMap Wiki.

[OpenStreetMap, 2014] OpenStreetMap, N. (2014). Import/Catalogue/N50 import (Norway) - OpenStreetMap Wiki.

[OpenStreetMap, 2016] OpenStreetMap, S. (2016). Stats - OpenStreetMap Wiki.

[OpenStreetMap, 2007] OpenStreetMap, T. (2007). TIGER - OpenStreetMap Wiki.

[OSM Tasking Manager,] OSM Tasking Manager, L. B. OSM Tasking Manager - LA buildings import.

[OSMF,] OSMF. About | OpenStreetMap Blog.

[OSMF, 2015] OSMF (2015). Membership/Statistics - OpenStreetMap Foundation Wiki.

[Palen et al., 2015] Palen, L., Soden, R., Anderson, T. J., and Barrenechea, M. (2015). Success & Scale in a Data-Producing Organization. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 4113–4122.

[Sambale, 2016] Sambale, M. (2016). OpenStreetMap | manings sin dagbok | Fixing split buildings in LA.

[Schleuss et al., 2016] Schleuss, J., Ureta, O., McConchie, A., and Sambale, M. (2016). Let's get LA on the map!: The Los Angeles Building Import Case Study | Seattle, Washington | State of the Map US 2016.

[Skogseth and Norberg, 2014] Skogseth, T. and Norberg, D. (2014). *Grunnleggende landmåling*. 1 edition.

[Wang et al., 2013] Wang, M., Li, Q., Hu, Q., and Zhou, M. (2013). Quality analysis of open street map data. *8th International Symposium on Spatial Data Quality*, 40(June):155–158.

[Willis, 2007] Willis, N. (2007). OpenStreetMap project imports US government maps | Linux.com | The source for Linux information.

[Willis, 2008] Willis, N. (2008). OpenStreetMap project completes import of United States TIGER data | Linux.com | The source for Linux information.

[Zielstra et al., 2013] Zielstra, D., Hochmair, H. H., and Neis, P. (2013). Assessing the effect of data imports on the completeness of openstreetmap - A United States case study. *Transactions in GIS*, 17(3):315–334.