

OpenStreetMap

Harpo 't Hart

Media Technology, Leiden
University
harpo_t_hart@hotmail.com

Wouter van den Heuvel

Media Technology, Leiden
University
w.vandenheuvel@gmail.com

Arjen Suijker

Media Technology, Leiden
University
arjensuijker@gmail.com

ABSTRACT

OpenStreetMap (OSM) is a worldwide virtual map that is freely accessible and editable by anyone, making it perfect for all kinds of projects that deal with geographical data. Its license is very open, which makes it a very interesting technology for creative expression. This report first informs the reader about the context of the OpenStreetMap project, explaining what it actually is and how it came to be. It goes on to explain the basic inner workings and then shows what the actual value of OSM is, making apparent what possibilities it offers. After this, the report concludes with a brief step-by-step tutorial to get the started with using the OpenStreetMap API through simple JavaScript.

1. PURPOSE, CONTEXT AND HISTORY

Purpose

OSM is an open source geographic data service. This means that anybody can contribute to the map at any time. The OpenStreetMap Foundation (OSMF) was founded because most maps that appear to be free actually have legal or technical restrictions on their use, holding back people from using them in creative, productive, or unexpected ways.[1] The OSMF gathers their data from different sources, including data-sets from governmental institutions, infrastructure companies, non-profit organizations and many individual users. [2]

OSM can be used by anyone, for both commercial and non-commercial use. The most common usages include:

- GPS routing software
- Collections of interesting geographic data
- Gaming
- Art

But because of the open license, many more possible implementations exist. In short, OSM provides accurate and extensive map data to the world, allowing everybody to do with it whatever they want.[3]

History

- 2004: The OpenStreetMap project was started in the UK. The focus of the project was mostly on gathering data on a national scale. Almost all of the data at this stage was collected by incorporating several large data-sets into one big digital map.

- 2006: Yahoo gives permission to use their aerial photography, making the maps much more user-friendly and accessible. A lot of effort goes into making it easier for users to add data to the map, so it becomes more accessible to less tech-savvy users.
- 2007: A complete data-set of all dutch roads and streets is donated by a company called Automotive Navigation Data, instantly making the map of the Netherlands accurate and useable for a lot of applications.[4]
- 2008: The OSM data-set is complete enough to replace proprietary geographic data on several types of GPS units, replacing their outdated data.
- 2013: The OpenStreetMap project welcomes their 1 millionth user. [5]



Illustration 1: Registered users over the years

2. OPERATING PRINCIPLES

Viewing maps

The output of OpenStreetMap is not visual, but rather the raw geographic data in XML format. XML is a very open, cross-platform language, which makes the data accessible on almost all computing devices. This data can be used in a variety of ways, but the most obvious one is to generate, (ie: render) map images. A plethora of renderers is available [6], facilitating different methods of rendering. The most common method used is *tile rendering*, where map visuals are created as a number of bitmap images, typically with a resolution of 256x256 pixels, to be positioned in a grid. [7]

The client then has a system in place to provide panning and zooming functionality. These actions cause the system to request downloading of new tiles as necessary to show the user the new areas of the map. This is also called a Slippy Map [8].

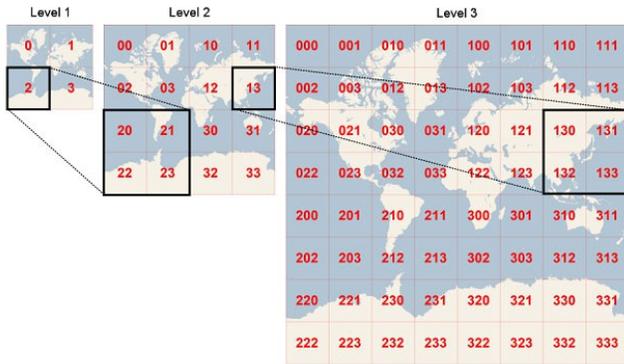


Illustration 2: Illustration of the tiled grid system. Image courtesy of Microsoft.

For each zoom level new tiles will have to be generated, this requires a substantial amount of computing power from the server, as well as data traffic. This is why smart caching mechanisms are usually in place on the server as well as the client, so that not all data will have to be resent every single time. However, most map providers implementing OSM data update their data extremely frequently. It is not uncommon to see map changes in effect globally within a few minutes.

The map on the OpenStreetMap website is a JavaScript powered web map. The tiles are generated using the Mapnik [9] toolkit, which is written in C++ and Python. There are also options available that allow for the usage of OpenCycleMap (which shows bicycle routes), public transport overlays and MapQuest [10] tiles.

The other method of rendering is direct vector rendering. This method translates the map data directly into graphics (vectors). This method only needs the raw data, which makes it much more data-efficient. The drawback is that it requires much more computing power from the client side. Vector rendering allows for several useful implementations, such as offline renderers, 3d-renderers, Flash renderers[10] and HTML5 renderers such as Cartagen[11] and Leaflet[12]. This is just a shortlist of the many implementations that are being created, most of which are interesting and cutting edge enough to warrant their own Web Technology Report.

Editing map data

Arguably the most interesting aspect of OSM is the fact that users, that is: anyone with a OSM account can make changes in the map data. Three data primitives can be edited: nodes, ways, relations. All of these can have tags associated with them.

- A **node** is a single geographical point using latitude, longitude and optionally altitude [13]. Nodes can represent anything, such as points of interest or mountain peaks. This is done through the use of tags.
- **Ways** are ordered collections of linked nodes [14]. Ways can consist of 2 to 2000 nodes. If the first node is connected to the last node it is called a closed way. Closed ways can be used to describe an area, such as forests, parking spaces or buildings.
- **Relations** are used to create logical or geographic relationships between nodes or ways [15]. It can be used, for example, to create routes or boundaries.

Aforementioned elements can be decorated with key-value pairs called tags to store metadata [16]. Tags provide semantics to OSM. Users are free to use whichever tags they deem most appropriate, but an extensive ontology of recommended tags exists [17].

Editing methods

In order to edit map data, one has to use editing software. Two varieties of such software exist:

- Online editing tools which run in the browser. Traditionally the tool of choice has been Potlatch and Potlatch2, [18] which runs in Flash. Recently, MapBox.com launched an alternative option called iD [19]. This editor is built in Javascript and aims to be fast and intuitive. The obvious benefit of such web based tools is that they are accessible from anywhere and there is a low threshold for new users to start editing.
- Offline editing tools are used often by more experienced users. These applications are installed on the computer and generally offer more features for power users. Two of the most used editors are Java based JOSM [20] and Merkaartor [21], which is written in C++.

Even more advanced users might want to opt to use the API directly. This API is used by the editors above and provides a complete interface to the entire database. For most other purposes it is too low-level to use. We will cover more about the API in the chapter 'Getting Started'.

3. STRENGTHS AND WEAKNESSES

To make a useful comparison between OSM and a number of the most popular web map services today (Google Maps, Yahoo! Maps, Bing Maps and MapQuest) one can look at a

number of criteria.

- The **license** under which the service is offered. OSM operates through a license model called Open Database License (ODbL) which permits distribution, as long as the same license is maintained.[22] This means that users can also charge money with anything they make using OSM. The other aforementioned mapping services are offered through proprietary licenses that are far more limiting. More important, these closed and proprietary platforms also do not offer the option for users to send in corrections or new data.
- **Map accuracy** in OSM is very high. OSM is based on a wikipedia-like system of user generated content that has resulted in a surprisingly accurate map. Helbrich et al [23] and Haklay [24] did an extensive research into what's referred to as Volunteered Geographic Information (VGI). They found that OSM has a high locational accuracy. They did also point out however, that there is some difference in the data density and quality between urban areas and the more rural parts. A possible weakness here can be data quality, because anyone is free to change or add data the risk exists that incorrect data enters the system. However a plethora [24] of quality assurance tools for the OSM data are in place. Furthermore, OSM data is updated 'on-the-fly' whereas other mapping tools have a turnaround of at least one year. Yahoo! Maps' data is at least four years old.
- **Coverage** is largest in OSM, all countries in the world are included in the data.
- **Offline use** is readily available (through for example OsmAnd). The complete map can be downloaded and used anywhere without an internet connection. Google maps also offers offline access, but this is limited to a small predefined area.
- **The amount of Zoom levels** OSM's tile renderer offers is 19, which is the same amount as Google Maps, but more than MapQuest (17). However, Yahoo! Maps provides 23 zoom levels, which allows it to give a more detailed view.
- **3D mode** is offered for OSM through osm-3d.org, besides OSM, Google Maps is the only service that provides a 3d view.
- **Printing maps** is another strength of OSM. A large number of options and possibilities exist [25]

Concluding, the major difference between OSM and other mapping web services is the fact that OSM is completely

open source and available for editing to everyone. This has both pro's and con's. As stated on the OSM FAQ [26], "By the very nature of the wiki-style process there is no guarantee of accuracy of any kind." This puts more responsibility on users of the map to verify the accuracy of the map. But since anyone can edit any map, anyone can also help improve its accuracy. So this is both a strength and a weakness. Websites such as GeoFabrik [27] can offer a side-by-side comparison between OSM and other maps.

4. INTENDED APPLICATIONS

The mission statement of OSMF states that the objectives of the OpenStreetMap project are among others: To make OSM data available for anyone (for editing and in whole) and that the data approaches geographic truth (or "ground truth"). [28] So the intended applications will be those that allow people to freely use and contribute to the maps.

There are lots of developers that use the geographical data of OSM for a wide range of applications, ranging from routing services to tools for rendering specialized maps to apps for editing the OSM geographical data [29]. The most common applications can be divided in four categories:

1. **Specialized maps or layers over existing maps** for a specific purpose or a specific group of people. For instance maps that show locations of special interest for queers [30], maps that show historical sites all over the world[31], or a map that shows the speed limits on the highways and roads in the Netherlands [32].
2. **Routing**, for instance a routing service for cyclists[33], or OpenTripPlanner[34], a multi-modal router combining OSM and Public transport tables.
3. Apps that **render maps that look pretty**, for instance in 3D[35], or maps that look like printed maps [36].
4. Apps for **editing the OSM data** and checking for errors. For instance an application that can show you the history of edits done in OSM[37]

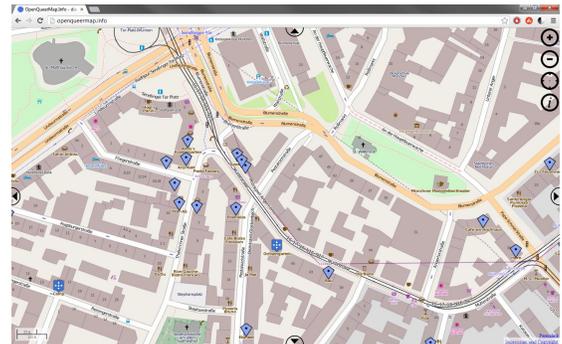


Illustration 3: OpenQueerMap

5. UNINTENDED APPLICATIONS

One could argue that given the OSMF mission statement the only unintended use of OSM is when you restrict people from freely accessing the OSM data or when you purposely submit false geographical data. This, however, is not a very interesting definition of 'unintended' applications. Therefore we decided to discuss implementations of the OSM data that aren't using the geographical data to gain knowledge about the world, but for other purposes.

Instances of this can be roughly divided in the following categories:

- The use of OSM data to create **art**
- The use of OSM data to create **games**.

The projects that use OSM data to make art are very diverse. A nice example of art made with OSM is an animation created out of maps[38]. Some other projects render the maps in an artistic way, like 8bitCities[39].

The maps are quite often used to create games. These can be divided into two categories:

1. Games that use OSM data to create a virtual world in a game.
2. Games that are played in the real world using OSM data.

A few games that use OSM maps as a virtual world are: FlightGear[40], a flight-simulator that allows you to fly over the map of the world, and Monopoly City Streets[41], a game of monopoly that allows you to buy actual streets of actual cities.



Illustration 4: Flightgear

In most of the games in the real world that use OSM, the player has to physically visit a location pointed on the map to complete a task. This is for instance the case with the

game Munzee[42], where you have to go to certain locations to scan a QR code that will award you points, or give you a task to complete in order to score points. Another nice example is the game Address Hunter[43]. In this game you have to go to a certain address, take a picture

and send the GPS-coordinates. This data is used by OSMF to verify the geographical data in their maps.

6. GETTING STARTED

This section will provide a brief instruction on how to work with the API (Application Programming Interface). The current version is API v0.6 [44], which provides easy access to the database.

This API can be accessed by a myriad of different technologies. This guide will focus on using JavaScript embedded in a web page to gather data from the OSM API. Following the instructions, you will create a web page that lists all the pubs in the center of Leiden. We assume a basic understanding of HTML and JavaScript. If you lack this knowledge, we suggest using w3schools.com as a helping hand throughout this tutorial.[45]

Creating a basic web page with jQuery functionality

Although this can also be done in basic JavaScript, we use a JavaScript library called jQuery that makes it all a little easier.[46] Go to <http://code.jquery.com/jquery-1.10.1.min.js> to download it. Then create a basic HTML file and include the jQuery library like so:

```
<!DOCTYPE HTML>
<HTML>
  <head>
    <script src='jquery-1.10.1.min.js' ></script>
  </head>
  <body>
  </body>
</HTML>
```

Getting the data

Now it is time to write some JavaScript. We start by making a jQuery get request [47] to the OSM API to get all the data within our coordinates:

```
<script>
$.get('http://api.openstreetmap.org/api/0.6/map?
bbox=4.47959,52.15412,4.49791,52.16421',
function(data) {
  alert('Data received!');
});
</script>
```

If you load this page and wait for a while, you should get a message telling you that you have received the data. As you can see, the API resides at <http://api.openstreetmap.org/api/0.6/map>. But because it's not very wise to try to download the full map of the world, we gave it some coordinates. These are included in the URL in the form of the variable 'bbox', which stands for 'bounding box'. You format the value as four numbers with comma's in between. The first is the longitude of the left border, the second is the latitude of the bottom border, the third is the longitude of the right border and the fourth is the latitude of the top border.

Finding the pubs

The data you receive is in XML format, and includes everything OSM knows about the area. Of course this is way more information than we need, so we have to filter this information. Things like pubs are always stored as 'nodes'. But these 'nodes' mean very little by themselves, they need 'tags' inside of them that describe what they mean. In XML terminology, a node is called the 'parent' of a tag. A simplified node would look like this:

```
<node>
  <tag k="amenity" v="pub"/>
  <tag k="name" v="Annie's"/>
  <tag k="URL" v="www.annies.nu"/>
</node>
```

The tags all have a 'k' and a 'v' attribute, which stands for a property-value pair. The first thing we will do is find all tags of which the amenity is pub. This goes between the curly brackets after `function(data)`, which we just created.

```
$(data).find("tag[v='pub']").each(
  function(){
    alert('Pub found!');
  }
);
```

This script looks for every 'tag' element of which the 'v' (value) attribute is 'pub', within the data we just requested. [48,49] Every time it finds one, it means that the node in which the tag is found (it's parent) must be a pub, so it creates a pop-up to tell you so.

Displaying the pub information

We will now find the tags that give more information about the pub. So instead of alerting "Pub found!", we do the following:

```
$(this).parent().find("tag[k='name'],tag[k='URL'], tag[k='phone'] ").each(function(){
```

```
  document.write( $(this).attr('k')+'!'+$
  (this).attr('v')+'</br>');
});
document.write('</br>');
```

So how does it work? `$(this)` stands for the tag that was just found. First, you select it's parent, which is the node. After this, you look within this node for every tag that describes the name, URL or phone number of the node. Every time we find one, we write it's value to the web page. You now have an always up-to-date list of pubs in Leiden, complete with contact information! Of course this API offers much more possibilities, so we would encourage you to play around with it.

7. FINAL THOUGHTS

OSM is a very useful technology. It creates many opportunities for practical implementations. But the most interesting about OSM may be the opportunities it offers for creative exploitation. Because the license is so open, almost anything is possible. In this report, we could only show the tip of the iceberg, but that should already be enough to serve as fertile soil for great ideas.

REFERENCES

- 1.OSM wiki: Main page, http://wiki.openstreetmap.org/wiki/Main_Page
- 2.OSM wiki: About, <http://wiki.openstreetmap.org/wiki/About>
- 3.OSM wiki: Applications, <http://wiki.openstreetmap.org/wiki/Applications>
- 4.OSM wiki: AND Data, http://wiki.openstreetmap.org/wiki/AND_Data
- 5.OSM wiki: History of OpenStreetMap, http://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap
- 6.OSM wiki: Rendering, <http://wiki.openstreetmap.org/wiki/Renderers>
- 7.J van der Bie, M van der Mark, V Vijn "Looking @ Google Maps". Media Technology, Leiden University
- 8.OSM wiki: Slippy Map http://wiki.openstreetmap.org/wiki/Slippy_Map
- 9.<http://mapnik.org/>
- 10.<http://open.mapquest.com/>
- 11.<http://cartagen.org/>
- 12.<http://leafletjs.com/>
- 13.OSM wiki: Node, <http://wiki.openstreetmap.org/wiki/Node>

14. OSM wiki: Way,
<http://wiki.openstreetmap.org/wiki/Way>
15. OSM wiki: Relation,
<http://wiki.openstreetmap.org/wiki/Relation>
16. OSM wiki: Tags,
<http://wiki.openstreetmap.org/wiki/Tags>
17. OSM wiki: Map Features
http://wiki.openstreetmap.org/wiki/Map_Features
18. OSM wiki: Potlach 2,
<http://wiki.openstreetmap.org/wiki/Potlach2>
19. John Firebaugh, 'New Map Editor Launches on OpenStreetMap.org', <http://www.mapbox.com/blog/new-map-editor-launches-openstreetmap/>
20. <http://josm.openstreetmap.de/>
21. <http://merkaartor.be/>
22. OSM wiki: Copyright and License
<http://www.openstreetmap.org/copyright>
23. M Helbich, C Amelunxen, P Neis, A Zipf.
"Investigations on locational accuracy of volunteered geographic information using OpenStreetMap data"
GIScience 2010 Workshop, Zurich, 2010
24. M Haklay "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets" Environment and planning. B, Planning & design, 2010
25. OSM wiki: OSM on paper,
http://wiki.openstreetmap.org/wiki/OSM_on_Paper
26. OSM wiki FAQ:
How can a project like this create accurate maps,
http://wiki.openstreetmap.org/wiki/FAQ#How_can_a_project_like_this_create_accurate_maps
27. <http://tools.geofabrik.de/mc/>
28. OSM wiki: Mission Statement,
http://www.osmfoundation.org/wiki/Mission_Statement
29. OSM wiki: List of OSM based Services,
http://wiki.openstreetmap.org/wiki/List_of_OSM_based_Services
30. <http://openqueermap.info/>
31. http://geschichtskarten.openstreetmap.de/historische_objekte/
32. <http://maximumsnelheid.openstreetmap.nl>
33. <http://www.naviki.org/nl/naviki/start/>
34. <http://opentripplanner.nl/index.HTML#/>
35. <http://www.osm-3d.org>
36. <http://alpha.map1.eu/#zoom=5&lat=50.083&lon=14.41&layers=BT>
37. <http://osmhv.openstreetmap.de/index.jsp>
38. Patrick Jean, Motorville, <http://vimeo.com/62468031>
39. <http://8bitcity.com/map>
40. <http://www.flightgear.org/>
41. http://wiki.openstreetmap.org/wiki/Monopoly_City_Streets
42. <http://www.munzee.com/>
43. <http://www.addresshunter.net/>
44. OSM wiki: API v.06,
http://wiki.openstreetmap.org/wiki/API_v0.6
45. <http://w3schools.com>
46. <http://www.jquery.com>
47. <http://api.jquery.com/jquery.get/>
48. <http://api.jquery.com/jquery.each/>
49. <http://api.jquery.com/category/selectors/>