

Verwendung von OpenStreetMap-Daten für eigene Projekte

Am Beispiel des Campusplans des KIT

Studienarbeit

Karlsruher Institut für Technologie
Institut für Photogrammetrie und Fernerkundung

can. geod.

Daniel Gann

(MatrikelNr.: 1355328)

Aufgabenstellung und Betreuung: Dipl.-Ing. Florian Schmidt

Karlsruhe, den 20. Dezember 2010

Lizenz

Dieses Werk ist unter einem "Creative Commons Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 3.0 Deutschland Lizenzvertrag" lizenziert. Um die Lizenz anzusehen, gehen sie bitte zu <http://creativecommons.org/licenses/by-nc-nd/3.0/de/> oder schicken sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Inhaltsverzeichnis

1	Einführung	1
2	Das OpenStreetMap Projekt	2
2.1	Definition.....	2
2.2	Komponenten und Datenformat.....	3
3	Arbeiten mit OpenStreetMap	7
3.1	Kartengenerierung aus Originaldaten.....	7
3.2	Zugriff auf die OSM Rohdaten.....	12
3.3	Thematische Datenauswahl.....	17
3.4	Bearbeitung.....	19
3.5	Aktualisierung.....	20
4	Fallbeispiel Campusplan	22
4.1	Beitrag von OpenStreetMap.....	23
4.2	Kartenoverlays.....	25
5	Zusammenfassung und Ausblick	26
	Literaturverzeichnis	I
	Abkürzungsverzeichnis	II

1 Einführung

In den letzten Jahren hat die Nutzung von Geodaten auch für den privaten Gebrauch immer mehr zugenommen. Vorangetrieben vor allem durch die Entwicklung des sogenannten Web 2.0 (oder auch "Mitmach-Web") [O'Reilly2005]. Beispielsweise bietet das soziale Netzwerk Facebook¹ seit Oktober mit seinem Dienst "Places" für seine Nutzer die Möglichkeit ihren derzeitigen Standort auf der eigenen Facebook-Seite zu veröffentlichen². Aber auch bereits etablierte Anwendungen wie Google Maps³ oder Google Earth⁴ erfreuen sich immer größerer Popularität und werden von vielen Anwendern benutzt um die verschiedensten Arten von raumbezogenen Fragen zu beantworten. Viele Nutzer von Geodaten im Internet sind sich der damit einhergehenden Probleme nicht bewusst. So unterliegen die Kartendaten von Google aber auch von anderen Anbietern wie beispielsweise Yahoo⁵ strengen Lizenzbedingungen die nicht nur eine kommerzielle Nutzung dieser Daten unmöglich machen sondern oft auch den Einsatz für private Zwecke erheblich erschweren. Zudem werden Informationen zur Vollständigkeit und Korrektheit der Daten oft nicht publik gemacht.

Aus diesem Grund wurde bereits 2004 in Großbritannien ein Projekt namens OpenStreetMap (OSM) gestartet. Dieses Projekt verfolgt das Ziel eine freie Weltkarte zu erstellen. Die OSM-Daten stehen unter einer freien Lizenz und können somit von jedermann lizenzkostenfrei betrachtet, benutzt und weiterverarbeitet werden. Die Möglichkeit auf diese Weise an aktuelle, nicht durch lizenzrechtliche Bedingungen eingeschränkte Geodaten zu gelangen eröffnet ein weites Feld an Anwendungsmöglichkeiten für die Erstellung von Karten und Pläne für private und kommerzielle Zwecke. Im Rahmen der vorliegenden Arbeit wird das exemplarisch am Beispiel des KIT gezeigt.

Im Oktober 2009 fusionierte die Universität Karlsruhe und das Forschungszentrum Karlsruhe zum KIT, dem Karlsruher Institut für Technologie. Für die Lagepläne der beiden ehemaligen Institutionen ist bis heute noch keine zufriedenstellende Fusion erreicht worden. Der ehemalige Campusplan der Universität ist sowohl technisch als auch inhaltlich nicht mehr auf dem aktuellen Stand, wird aber Mangels brauchbarer Alternativen weiter betrieben. Für das ehemalige Forschungszentrum, heute KIT Campus Nord, existiert sogar nur die PDF Version eines analogen Plans. Diese Situation ist für alle Beteiligten äußerst unbefriedigend. Daher wurde schon in [Veit2010] dargelegt, welchen Anforderungen ein neuer Plan genügen muss. Als ein Ergebnis dieser Arbeit konnte festgehalten werden, dass es sinnvoll ist als Datengrundlage eines neuen Planes die Daten des OpenStreetMap-Projektes zu nutzen.

Im Rahmen dieser Arbeit wird dafür zunächst das OSM Projekt und sein technischer Unterbau sowie die verwendeten Datenstruktur näher erläutert. Anschließend wird dargestellt wie die Arbeit mit OSM Daten funktioniert, welchen technischen Eigenarten die verschiedenen Methoden unterliegen und ihre Vor- und Nachteile herausgearbeitet. Dies wird dann beispielhaft für den Campusplan des KIT gezeigt.

1 Facebook: <http://www.facebook.com/>

2 Facebook-Blog: <http://blog.facebook.com/blog.php?q=Facebook+Places>

3 Google Maps: <http://maps.google.de/>

4 Google Earth: <http://earth.google.de/>

5 Yahoo Maps: <http://maps.yahoo.com/>

2 Das OpenStreetMap Projekt

Wie in der Einleitung bereits erwähnt beschäftigt sich diese Arbeit intensiv mit dem Thema OpenStreetMap. Um dabei keine Unklarheiten aufkommen zu lassen, gibt dieses Kapitel einen Überblick über die wichtigsten Aspekte des OSM Projektes und der dahinter liegenden Technik die für das weitere Verständnis dieser Arbeit vonnöten sind.

2.1 Definition

OpenStreetMap ist ein im Jahr 2004 in Großbritannien ins Leben gerufenes Projekt das sich zum Ziel gesetzt hat “[...] eine freie, editierbare Karte der gesamten Welt, die von Menschen wie dir erstellt wird.[...]“ [OpenStreetMap] zu erschaffen. Dabei wird ein besonderes Augenmerk auf “frei“ gelegt das hier im in der Open-Source Bewegung üblichen Sinn verwendet wird. Der große Unterschied zwischen OSM und anderen Online Kartenangeboten ist nicht das OSM kostenlose Daten anbietet, sondern die Tatsache dass diese auch ohne technische Einschränkungen und unter einer freien Lizenz zur Verfügung gestellt werden. Die hierbei verwendete Lizenz ist die “Creative Commons Attribution-Share-Alike“ in der Version 2.0 (oder kurz: CC-BY-SA 2.0⁶) die besagt das die Daten beliebig vervielfältigt, verbreitet, weiterverarbeitet und öffentlich gemacht werden dürfen wen der Rechteinhaber genannt wird und ein aus den Daten erstelltes Werk ebenfalls unter der gleichen oder einer vergleichbaren Lizenz veröffentlicht wird

In vielerlei Hinsicht ähnelt das Projekt dabei der bekannten freien Enzyklopädie Wikipedia. Auch dort wird das “Mitmach-Web“ mit Erfolg gelebt. Jeder kann dort seinen Beitrag leisten indem er neue Informationen einbringt oder bestehende verbessert. OpenStreetMap verfolgt den gleichen Ansatz. Auch hier kann jeder mitmachen. Die einzige Voraussetzung ist die Erstellung eines kostenlosen Accounts, anschließend kann jeder Interessierte mit der Erfassung von ihn interessierenden Kartendaten beginnen.

OSM setzt dabei vor allem auf die Nutzung von GPS für die Erstellung ihrer Karten. Dabei werden die von GPS Geräten in regelmäßigen Abständen erzeugten Wegpunkte, sogenannten Trackpoints, benutzt um anhand dieser einen Straßenverlauf rekonstruieren zu können. Abbildung 2.1 zeigt die so entstandenen Tracks im Bereich des KIT Campus. Das Straßennetz ist hier deutlich zu erkennen. Die mithilfe dieser Tracks eingezeichneten Straßen werden dann “getaggt“ also mit Paaren von Schlüsseln und Werten, oder key-value Paaren, wie sie in der OSM Gemeinde üblicherweise genannt werden, versehen. Mithilfe dieser Tags werden die Eigenschaften des zu erfassenden Objekts beschrieben. Ob es sich also um eine Hauptstraße oder eine Fußgängerzone handelt, ob es eine Einbahnstraße ist oder ob ein Tempolimit vorgegeben ist. OSM lässt den Nutzern dabei völlige Freiheit was alles getaggt werden kann. Jeder kann neue Tags erfinden und benutzen, eine Liste mit den gebräuchlichsten und üblichen Tags findet sich im [OpenStreetMapWiki] unter dem Stichwort “Map_Features“.

6 CC-BY-SA 2.0: <http://creativecommons.org/licenses/by-sa/2.0/de/>

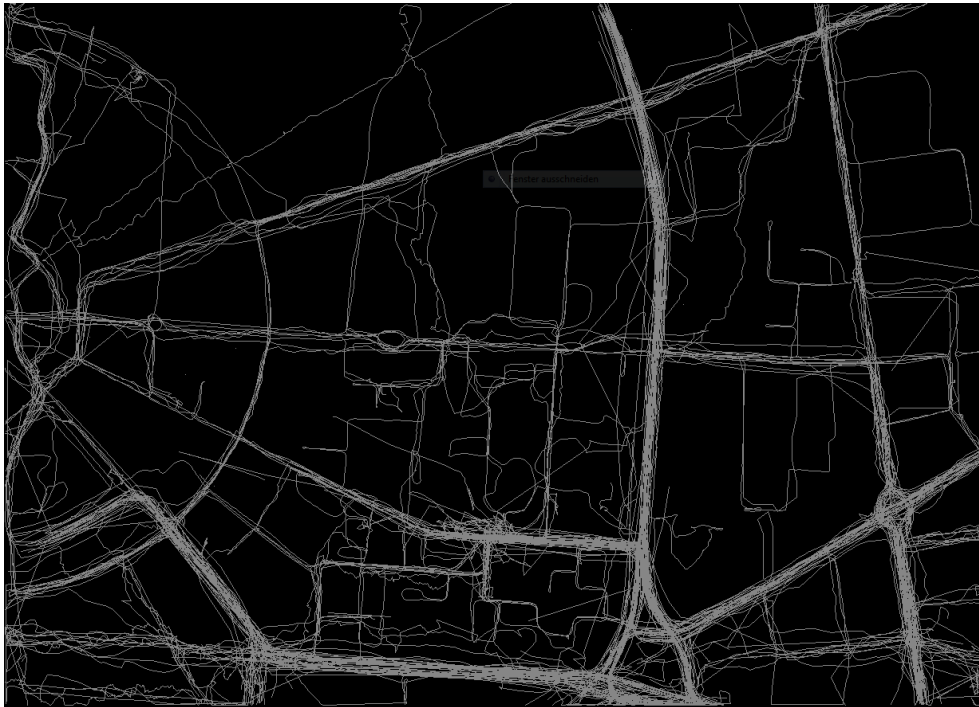


Abbildung 2.1: GPS Tracks in Bereich des Campus des KIT.

Aber nicht nur Straßen werden in OSM erfasst. Auch andere Objekte die von Interesse sein können, sogenannte Points of Interest (POI) werden erfasst und getaggt. So finden sich dann Polizeiwachen, Einkaufshäuser oder Spielplätze ebenfalls in den Daten wieder.

Aber nicht nur GPS wird als Datengrundlage verwendet. Auch frei nutzbare Luftbilder wie etwa die des NASA Satelliten Landsat oder die zum Abzeichnen freigegebenen Luftbilder von Yahoo können benutzt werden. Darüber hinaus wurden frei verfügbare Geodaten wie etwa die des US Amerikanischen TIGER Projekts⁷ oder des Frida Projekts⁸ von Osnabrück bereits in das OSM Datenmaterial integriert.

2.2 Komponenten und Datenformat

Grob vereinfacht lässt sich sagen, das OSM Projekt besteht aus vier Komponenten (Abbildung 2.2). Diese wären:

1. Editoren zur Bearbeitung der Daten
2. Datenbank für die Datenhaltung
3. Renderer zur Kartenerstellung
4. Viewer zur Kartenanzeige

Als Editoren stehen mehrere von der SM Gemeinde entwickelte Programme zur Verfügung. Die beiden verbreitetsten sind der auf Flash basierende Online Editor Potlatch und der in Java geschriebene Editor JOSM. Dabei besitzt JOSM eine Vielzahl von Einstellmöglichkeiten und

⁷ Projekt TIGER: <http://www.census.gov/geo/www/tiger/>

⁸ Frida Projekt Osnabrück: <http://frida.intevation.org/>

Zusatzprogrammen und ist daher eher auf die Bedürfnisse von erfahrenen Benutzern zugeschnitten während Potlatch nur die benötigten Basisfunktionalitäten besitzt und somit besser für Neueinsteiger geeignet ist.

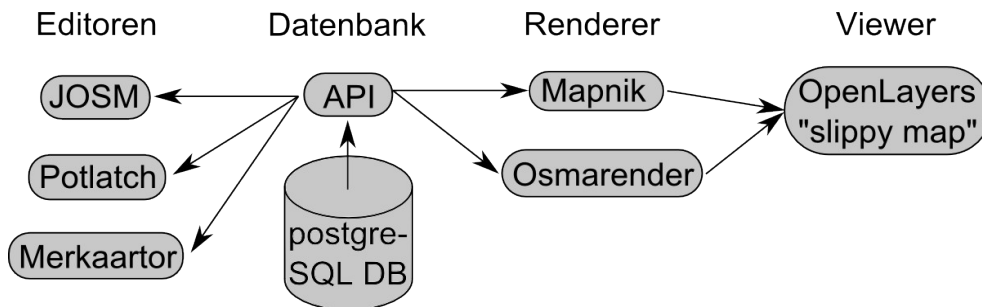


Abbildung 2.2: Übersicht über die Komponenten von OSM (Vereinfacht aus [OpenStreetMapWiki]).

Zur Datenspeicherung kommt ein Datenbankserver mit PostgreSQL zum Einsatz. Dieser, sowie alle weiteren für den Betrieb von OSM benötigten Server stehen am University College in London. Nach dem Zugriff auf die Daten durch das Application Programming Interface (API) liegen diese in einer .osm Datei vor. Diese ist eine speziell formatierte XML Datei in der die Geometrien und Attribute der in OSM erfassten Objekte gespeichert sind.

Zur Kartenerstellung benutzt OSM zwei verschiedene Renderer um aus den Geodaten eine darstellbare Karte zu erzeugen. Diese beiden sind Osmarender und Mapnik. Osmarender ist speziell für OSM entwickelt worden. Dabei war ausschlaggebend, dass die OSM Rohdaten in einem XML basierten Datenformat vorliegen und mit SVG ein Vektor-Grafikformat existiert das ebenfalls XML basiert ist. Weiterhin spielten die Möglichkeiten der Extensible Stylesheet Language Transformation (XSLT), die es erlauben XML Dokumente in andere XML Dokumente umzuwandeln, eine große Rolle. Osmarender selbst ist eine Sammlung von Dateien mithilfe derer es einem weiteren Programm (dem XSLT Prozessor) ermöglicht wird eine solche Transformation durchzuführen an deren Ende eine bildliche Darstellung der Geodaten steht. Mapnik hingegen ist eine in C++ geschriebene freie Werkzeugsammlung zum Rendern von Karten. Dafür benötigt Mapnik Zugriff auf eine Datenbank in der die Rohdaten gespeichert sind und erstellt daraus Rastergrafiken. Daher sind mit Mapnikkarten primär für die Verwendung am Bildschirm gedacht, während für Ausdrücke der Karten dank der verlustfreien Skalierbarkeit einer Vektorgraphik die Verwendung von Osmarender vorteilhaft ist.

Für die Darstellung der Karten wird eine Kombination einer WebMapService (WMS) Serverimplementierung und OpenLayers genutzt um im Hintergrund Kartenkacheln dynamisch vom Server abzurufen ohne die ganze Seite erneut laden zu müssen. Auf diese Weise ist ein gleichmäßiges Betrachten und ein schnelles Verschieben und Zoomen der Karte möglich. Daher rührt auch der in der OSM Gemeinde dafür verwendete Begriff "slippy map" der auf Deutsch flinke Karte bedeutet.

Datenformat

Das .osm Datenformat wurde nie formal spezifiziert. Daher ist es möglich das auf verschiedenen Wegen erstellte Dateien geringfügige Unterschiede aufweisen die jedoch für die meisten Anwendungszwecke vernachlässigbar sind. Im Grunde ist eine .osm Datei nichts anderes als eine Auflistung einer beliebigen Anzahl der drei verschiedenen grundlegenden Datentypen des OSM

Datenmodells (Abbildung 2.3). Diese sind Node, Way und Relation. Dabei besteht jedes Element aus einzelnen key-value Paaren, mit welchen den Objekten ihre Eigenschaften zugeschrieben werden. Diese werden im Tag-Element mit den beiden Werten k für key und v für value beschrieben. Identifiziert werden die Objekte über eine eindeutigen Kennziffer, den Identifikator (ID). Diese Kennung wird selbst nach dem Löschen des Objektes nicht erneut freigegeben. Dabei werden drei unterschiedliche Nummernräume benutzt, es kann also sowohl einen Node, einen Way als auch eine Relation mit der ID 1 geben. Wenn in einem der Editoren Änderungen gemacht werden, benutzen diese für neue Objekte temporäre negative Nummern die erst nach dem Einpflegen in die OSM Datenbank durch die endgültige ID ersetzt werden.

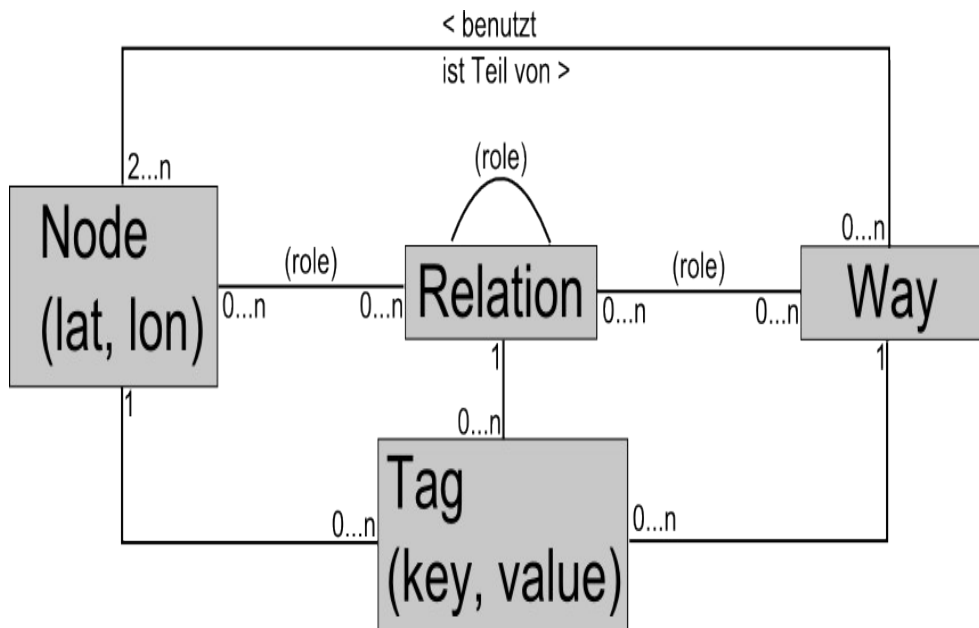


Abbildung 2.3: Vereinfachte Darstellung des OSM Datenmodells [Ramm2008].

Nodes sind dabei die Basiselemente. Jeder Node ist ein Punkt der neben der ID noch bekannte geographische Länge (lat) und Breite (lon) besitzen muss. Zur Genaueren Beschreibung kann er noch beliebig viele weitere key-value Paare besitzen. Üblicherweise werden mindestens der Name des Bearbeiters (user) und der Zeitpunkt der Bearbeitung (timestamp) mit angegeben.

Ways sind Linienzüge die aus 2 bis 2000 Nodes bestehen können welche durch Geraden verbunden werden. Wenn dabei der erste und der letzte Node identisch sind, also ein geschlossener Linienzug gebildet wird kann dieser zur Repräsentation einer Fläche genutzt werden da ein eigenes Element dafür im OSM Datenmodell nicht vorgesehen ist. An Tags erfordert jeder Way neben seiner ID noch die Kennungen der Nodes über die er sich definiert (nd ref). Darüber hinaus kann jeder Way noch eigene key-value Paare haben die nicht mit denen seiner Nodes übereinstimmen müssen.

Das letzte zentrale Element sind Relations. Eine Relation ist eine sortierte Liste von beliebig vielen Nodes, Ways oder auch anderen Relations die jeweils eine bestimmte Rolle übernehmen. Dafür wird neben der Kennung (ref) noch der Typ (Node, Way, Relation) des Objektes und seine Verwendung in der Relation (role) benötigt. Darüber hinaus kann auch eine Relation noch eine beliebige Anzahl von Tags besitzen. Ein Beispiel für eine Relation sind Abbiegevorschriften, die aus zwei Ways, einem für den Weg aus dem man heraus abbiegt, einem in den man abbiegen will, sowie einem Node an welchem die Vorschrift in Kraft tritt, bestehen. Ein Anderes Beispiel wäre eine

Straße die aus verschiedenen Ways besteht weil sie Abschnittsweise mit verschiedenen Attributen versehen werden soll, wie etwa verschiedenen Geschwindigkeitsbegrenzungen, einem Brücken- oder Tunnelabschnitt oder einer Fußgängerzone.

Die Einträge werden dabei nach Nodes, Ways und Relations sortiert in der .osm Datei aufgeführt. Das ganze wird von einem zentralen osm Element ummantelt in welchem weiterführende Informationen enthalten sind, etwa zur Version oder dem Verwendeten Programm. Hinzukommt ein Element das die Räumliche Ausdehnung des Bereiches für den Daten enthalten sind definiert. Eine .osm Datei würde also in etwa folgendermaßen aussehen:

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">
<bounds minlat="..." minlon="..." maxlat="..." maxlon="..."/>
  <node id="20246262" lat="..." lon="..." user="..." timestamp="..." ...>
    <tag k="..." v="..."/>
    ...
  </node>
  ...
  <way id="4689368" user="..." timestamp="..." ...>
    <nd ref="20246262"/>
    ...
    <tag k="..." v="..."/>
    ...
  </way>
  ...
  <relation id="939026" user="..." timestamp="..." ...>
    <member type="node" ref="20246262" role="..."/>
    ...
    <member type="way" ref="4689368" role="..."/>
    ...
    <member type="relation" ref="..." role="..."/>
    ...
    <tag k="..." v="..."/>
    ...
  </relation>
  ...
</osm>
```

3 Arbeiten mit OpenStreetMap

Nachdem das letzten Kapitel den Aufbau der OSM Daten näher erläuterte wird nun auf die verschiedenen Möglichkeiten eingegangen mittels derer die vorhandenen Daten des OSM Projektes für eigene Zwecke benutzt werden können. Angefangen mit den unterschiedlichen Zugriffsmöglichkeiten auf die Daten, den Wegen die Daten den eigenen Bedürfnissen entsprechend zu modifizieren und abschließend wie die Daten eines eigenen Projektes aktuell gehalten werden können.

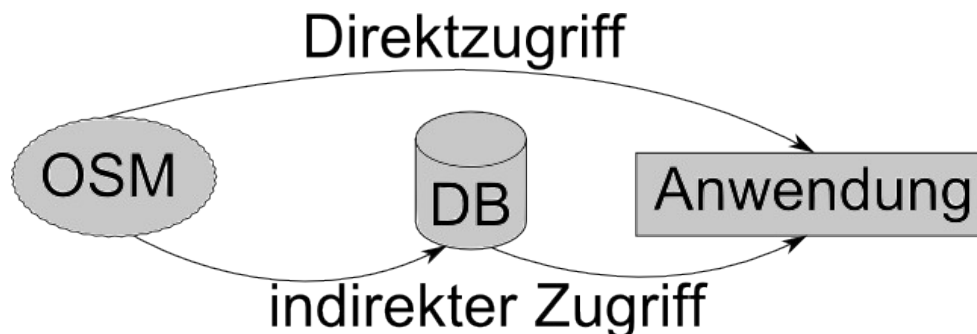


Abbildung 3.1: Verschiedene Möglichkeiten um mit den OSM Daten zu arbeiten.

Dafür wird zunächst einmal zwischen den beiden Möglichkeiten unterschieden entweder direkt mit den von OSM zur Verfügung gestellten Daten zu arbeiten ohne dabei die Möglichkeit zu haben diese bearbeiten zu können oder aber die Daten zunächst herunterzuladen, eventuell zu modifizieren und anschließend in eine eigene Datenbank zu laden und von dort aus für die Verwendung als Webdienst oder zur Kartenberechnung vorzuhalten. In Abbildung 3.1 sind diese zwei Grundsätzlichen Möglichkeiten des Zugriffs auf die Daten simplifiziert dargestellt.

3.1 Kartengenerierung aus Originaldaten

Für einfache Projekte kann es ausreichend sein direkt mit den in OSM verfügbaren Daten zu arbeiten. Dies ist immer dann der Fall wenn die auf der OSM Hauptseite dargestellte Version bereits alle relevanten Daten enthält und somit keine weitere Bearbeitung der Daten nötig ist. Dabei muss beachtet werden das dabei nur die OSM Daten für die Ansicht genutzt werden und keine eigenständige Karte generiert wird. Daher können auch nur die von OSM angebotenen Kartenstile und Informationen benutzt werden.

Die einfachste Möglichkeit sich auf diese Art eine Karte zu generieren ist der sogenannte Permalink, der mit einer OSM Ansicht verknüpft ist und benutzt werden kann um diese später erneut aufzurufen. Alternativ kann die OSM Karte auch in eine eigene Webseite eingebunden

werden. Dafür wird eine JavaScript Bibliothek benötigt welche die notwendigen Funktionen hierfür bereitstellt. Die am weitesten verbreitete ist OpenLayers⁹ aber auch die Einbindung mittels Mapstraction¹⁰ oder der CloudMade Web Maps API¹¹ ist möglich.

Permalink

Unter der Adresse des Permalinks sind die Daten der momentanen Ansicht hinterlegt und können so für eine spätere Verwendung oder zur Weitergabe gespeichert werden. Um sich also beispielsweise für eine Wegbeschreibung eine Karte zu besorgen genügt es in der OSM Ansicht zur entsprechenden Position zu navigieren und den in der rechten unteren Ecke eingeblendeten Link zu benutzen. Neben der aktuellen Position wird auch der verwendete Kartenstil als Parameter benutzt. Durch das Einfügen der Parameter `mlon` und `mlat` in die URL kann zusätzlich eine Markierung an der durch die Parameter spezifizierten Stelle erstellt werden. Neben der Darstellung des Ausschnitts über den Kartenmittelpunkt ist auch eine Angabe über eine Bounding Box (BBox) möglich. Um hierbei allerdings verzerrte Darstellungen zu vermeiden erfolgt die Darstellung in einem vom Programm vorgegebenen Seitenverhältnis. Es kann also möglich sein, dass Teile der BBox außerhalb des sichtbaren Ausschnittes liegen. Eine Übersicht aller Parameter gibt Tabelle 3.1

Tabelle 3.1: Die Parameter des OSM-Permalinks.

Parameter	Erklärung	Inhalt
lat	Breite der Kartenmitte	Geographische Breite in Dezimalzahlen mit “.” als Trennzeichen. Positiv für nördliche, negativ für südliche Koordinaten
lon	Länge der Kartenmitte	Geographische Länge in Dezimalzahlen mit “.” als Trennzeichen. Positiv für östliche, negativ für westliche Koordinaten
minlat	Breite der unteren Kante	Alternative Angabe ; Format entsprechend lat
maxlat	Breite der oberen Kante	
minlon	Länge der linken Kante	Alternative Angabe; Format entsprechend lon
maxlon	Länge der rechten Kante	
mlat	Breite der Markierung	Optionale Angabe, Format entsprechend lat
mlon	Länge der Markierung	Optionale Angabe, Format entsprechend lon
zoom	Zoom-ebene	Von 0 (ganze Welt) bis 18 (Detailansicht)
layers	Kartendarstellung	Code für die Darstellungsart

Mit dem Code für die Darstellungsart kann neben der Auswahl des Kartenstils auch das von OSM bereitgestellte Datenoverlay angezeigt werden. Details zum Aufbau des Codes liefert Tabelle 3.2. Dabei ist der an vierter Stelle stehende Codebuchstabe (F bzw. T) momentan funktionslos, da das über ihn angesprochene Qualitätssicherungstools Maplint im März 2010 eingestellt wurde.

⁹ OpenLayers: <http://openlayers.org/>

¹⁰ Mapstraction: <http://www.mapstraction.com/>

¹¹ CloudMade: <http://cloudmade.com/>

Dafür bietet OSM mittlerweile auch eine vierte Darstellungsart an bei der Straßen ohne Namen speziell gekennzeichnet sind um diese einfacher finden zu können und das aktualisieren zu erleichtern. Dieser ist durch den Code in dieser Form nicht ansprechbar. Es können zwar die ersten 3 Buchstaben des Codes durch ein "N" ersetzt werden um so die Ansicht aufzurufen. Allerdings sind dann die durch die übrigen 3 Buchstaben angesprochenen Funktionen nicht mehr aufrufbar. Jedoch ist dies auch bei den alternativen Codedarstellungen für die anderen Kartenstile der Fall.

Tabelle 3.2: Code für die Kartendarstellung des Permalinks [OpenStreetMapWiki].

Code	Kartendarstellung
B 0 0	Darstellung durch Mapnik; alternativ auch "M" möglich
0 B 0	Darstellung durch Osmarender; alternativ auch "O" möglich
0 0 B	Darstellung durch Cycle Map; alternativ auch "C" möglich
F T F	Darstellung ohne zusätzliche Anzeige Ebene
F T T	Kombiniert mit Datenoverlay

Die so entstehenden Links sind allerdings oft sehr lang, wodurch die Handhabung dieser Links erschwert wird. Zur Vereinfachung bietet OSM neben dem Permalink noch einen "Shortlink" an. Dabei werden die in der URL enthaltenen Informationen kodiert und somit stark verkürzt. Beispielsweise verweisen die beiden folgenden Links auf den selben Kartenausschnitt.

<http://www.openstreetmap.org/?lat=49.0159&lon=8.4117&zoom=13&layers=B00FTF>

<http://osm.org/go/ODPvniS>

Dadurch sind die Parameter jedoch nicht mehr vom Benutzer lesbar und eine Anpassung beispielsweise durch Einfügen einer Markierung ist nicht mehr möglich.

Einbindung in eigene Website mit OpenLayers

Wenn die OSM Karte nicht als Link benutzt werden soll kann sie auch mit OpenLayers als Element in eine Website integriert werden. Dazu muss lediglich das OpenLayers Script in die Website eingebettet werden. Ob dabei das Script von der OpenLayers Seite verlinkt wird oder eine Kopie davon auf dem eigenen Server liegt spielt dabei keine Rolle. Nach der Verlinkung bietet OpenLayers Funktionen an mithilfe derer die verschiedenen OSM Layer in eine OpenLayers Karte eingebunden werden können. OpenLayers stellt dazu noch etliche weitere Funktionen für die Anpassung der Karte zur Verfügung. So gibt es die Möglichkeit verschiedene Kontrollelemente hinzuzufügen oder einen bestimmten Kartenausschnitt nach dem Laden anzuzeigen. Ein Beispiel dafür findet sich auf der Webseite zu [Ramm2008]¹² (Abbildung 3.2). Weiterführende Informationen zum Thema Kartenerstellung mit OpenLayers bietet [Jansen2010].

¹² Webseite zum Buch unter: <http://www.openstreetmap.info>

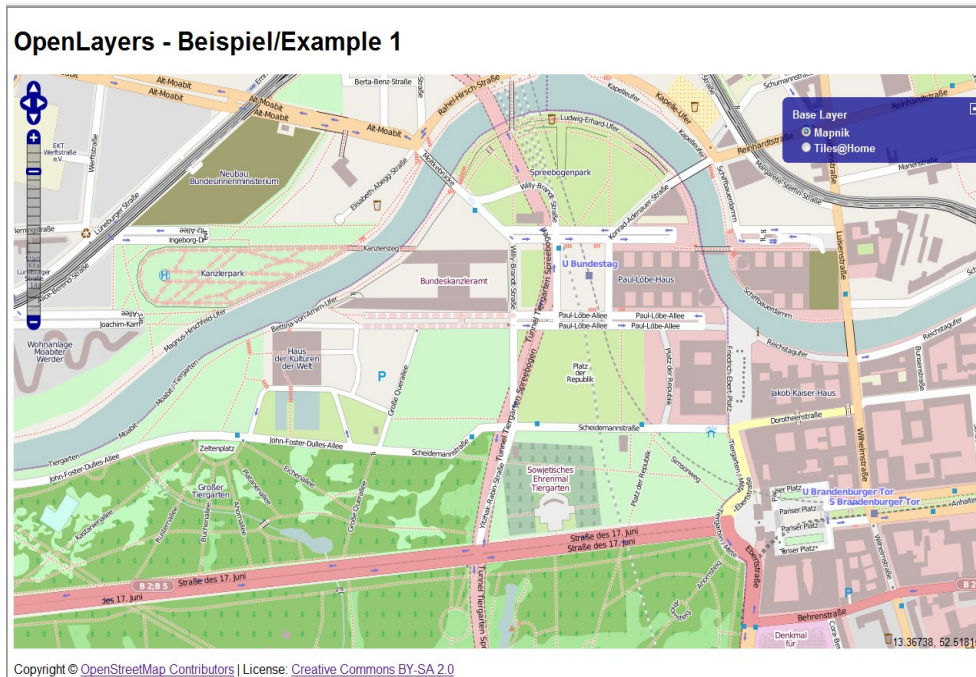


Abbildung 3.2: OpenLayers mit OpenStreetMap-Daten für Berlin-Mitte.

<http://www.openstreetmap.info/examples/auflage3-ol-example1.html>

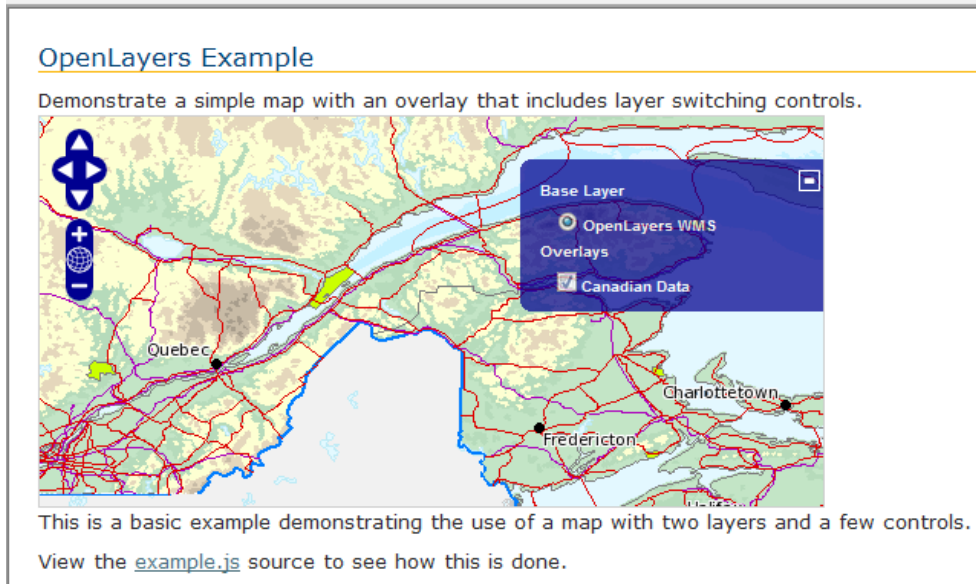


Abbildung 3.3: OpenLayers mit Kanadischem Datenoverlay für die Umgebung von Quebec.

<http://www.openlayers.org/dev/examples/example.html>

3.1 Kartengenerierung aus Originaldaten

Es besteht auch die Möglichkeit außer den OSM Layern noch weitere Layer aus anderen Quellen einzubinden. So können beispielsweise die Layer von Google Maps mit eingebunden werden. Voraussetzung dafür ist die Registration bei Google um einen, für die Nutzung der Google-API benötigten, Schlüssel zu erhalten. Das Einfügen von Layern anderer Kartenanbieter wie Yahoo Maps oder Bing Maps¹³ ist ebenfalls möglich.

Aber nicht nur Kartenlayer lassen sich in OpenLayers einbinden. Stehen zusätzliche Daten zur Verfügung, beispielsweise in Form eines WFS Servers, können auch diese in Form eines Overlays über die Karte gelegt werden. Ein Beispiel dazu findet sich auf der OpenLayers Seite (Abbildung 3.3). Die Möglichkeit als Datenoverlay eingebunden zu werden existiert auch für ArcGIS Server-, KML- und GML-Daten sowie einige weitere Formate.

Um die Erstellung solcher Karten für die private Nutzung zu vereinfachen wurde 2009 von Sebastian Hohmann ein OSM Online-Kartengenerator namens "Easymap"¹⁴ entwickelt. Dieses Tool vereinfacht die Erstellung einer auf OpenLayers basierenden Karte. Statt die für die Karte benötigten Funktionen selbst händisch in den Quelltext einzuarbeiten muss man hier lediglich die gewünschten Elemente aus einer Liste auswählen, den korrespondierenden Code erzeugt das Tool dann selbstständig. Der gewünschte Kartenausschnitt kann auf einer eingebauten Karte entweder selbst angefahren werden oder, wenn die Parameter des Kartenmittelpunktes bekannt sind, gibt man die diese direkt in die Eingabefelder ein. Auch eine Funktion zum Setzen von Markierungen zusammen mit einer kurzen Textbeschreibung ist vorhanden. Als besonderes Feature können auch Polygone und Linien interaktiv in der eingebauten Karte eingezeichnet und als Overlay über die fertige Karte gelegt werden. Die Farben, Füllungen und Beschreibungen der Zeichnungen sind dabei ebenfalls den individuellen Wünschen anpassbar. Abbildung 3.4 zeigt Exemplarisch wie eine solche Karte aussehen kann.



Abbildung 3.4: Beispiel für eine mit Easymap erstellte Karte.

13 Bing Maps: <http://www.bing.com/maps/>

14 Easymap: <http://osmtools.de/easymap/>

Allerdings ist Easymap eher für kleine Projekte geeignet. Für große Projekte sind die Möglichkeiten des Tools nicht umfangreich genug und können höchstens als Ausgangsbasis oder zur Inspiration dienen, Denn das Hinzufügen von Markierungen und Zeichnungen ist aufwändig und kann nur bei der Erstellung der Karte geändert werden. Wenn zu einem späteren Zeitpunkt etwa eine zusätzliche Markierung hinzugefügt werden soll, müssen alle Einstellungen und bereits gesetzte Marker und Zeichnungen erneut in den Generator eingegeben werden. Auch ist eine Individualisierung was die Kartendarstellung an sich betrifft nicht möglich, da dafür auf die von OSM bereitgestellten Möglichkeiten zurückgegriffen wird.

3.2 Zugriff auf die OSM Rohdaten

Für große Projekte sind die Daten wie sie von OSM bereitgestellt werden nicht immer ausreichend. Viele Arbeitsschritte werden ungleich einfacher wenn man mit den Rohdaten der Karten anstatt der schon fertigen Karten arbeiten kann. Um mit den Rohdaten des OSM Projektes arbeiten zu können benötigt man in geeigneter Weise Zugriff auf dieses Datenmaterial.

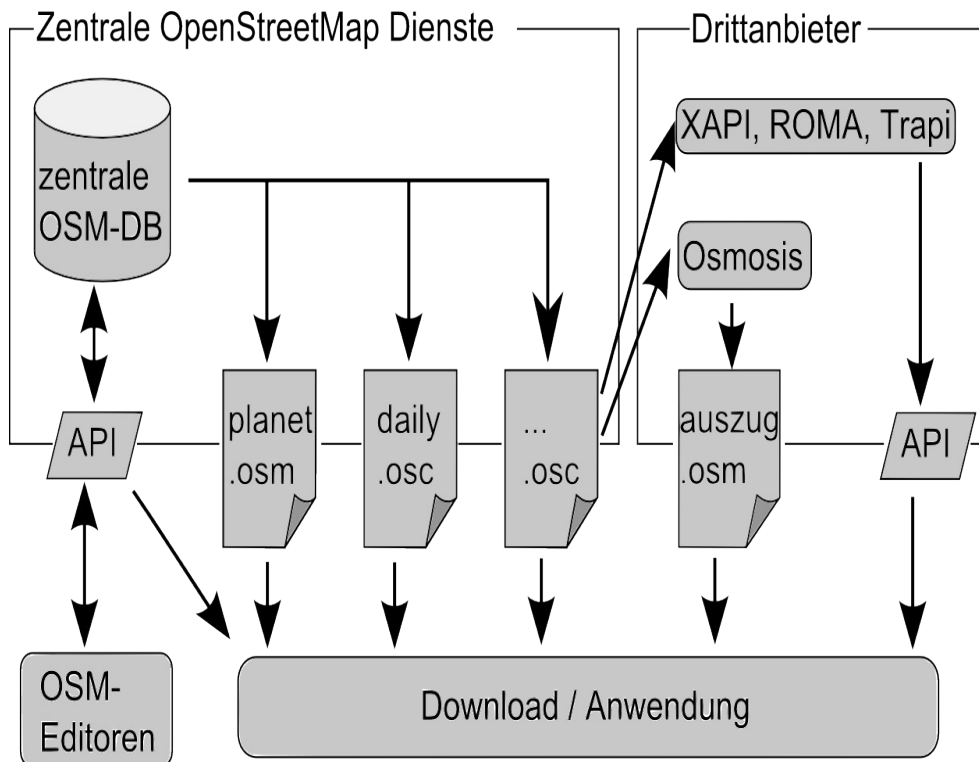


Abbildung 3.5: Verschiedene Wege um auf die OSM Rohdaten zuzugreifen [Ramm2008].

Die verschiedenen Ansätze für den Zugriff sind in Abbildung 3.5 dargestellt. Um die Daten eines ausgewählten Bereiches zu erhalten kann zum einen die OSM API benutzt werden. So erhält man immer die aktuellen Daten. Allerdings kann über die API nur ein kleiner Ausschnitt aus der Datenbank heruntergeladen werden. Wird ein größerer Bereich benötigt besteht die Möglichkeit einen kompletten Datenbank Auszug zu nehmen und sich den den gewünschten Bereich selbst zu extrahieren. Oder man benutzt Drittanbieter die ihrerseits die Datenbank Auszüge verwenden um gefilterte, ausgeschnittene oder auf andere Weise vor verarbeitete Auszüge bereitstellen. Ein

Beispiel für einen solchen Drittanbieter ist die Karlsruher Firma Geofabrik¹⁵ die auf ihrer Seite vorgefertigte Ausschnitte zum Download anbietet. Andere Drittanbieter bieten statt einem Download Dienst auch eigene API an über die dann auf deren Daten zugegriffen werden kann.

Die OSM API

Der Zugriff auf OSM Daten über die API erfolgt über folgende HTTP GET Anweisung die an den API-Server¹⁶ gestellt wird:

```
GET /api/0.6/map?bbox=left,bottom,right,top
```

Dabei entsprechen die vier Variablen left, bottom, right und top den BBox Parametern minlon, minlat, maxlon und maxlat die schon in Tabelle 3.1 näher erläutert wurden. Daraus ergibt sich dann beispielsweise eine solche URL.

```
http://api.openstreetmap.org/api/0.6/map?bbox=13.33,52.51,13.38,52.52
```

Diese Anfrage liefert in etwa das Areal des Tiergartens in Berlin. Dabei erfolgt die Antwort der API als XML Dokument das der OSM Datenstruktur entspricht und daher problemlos als .osm Datei gespeichert und behandelt werden kann.

An Daten enthält das XML Dokument folgendes

- alle Nodes welche sich im durch die BBox begrenzten Bereich befinden
- alle Ways bei denen wenigstens ein Node im Bereich liegt
- alle weiteren Nodes die zu solch einem Way gehören auch wenn diese außerhalb liegen
- alle Relations die sich auf wenigstens einen Node oder Way innerhalb der BBox beziehen.

In Abbildung 3.6 ist deutlich zu sehen wie einzelne Wege über das ausgewählte Gebiet hinausreichen. Relations werden im Gegensatz zu Ways allerdings nicht rekursiv behandelt. Um übergroße Datenmengen zu vermeiden werden nur diejenigen Nodes und Ways einer Relation mitgeliefert die auch auch im Gebiet liegen. Ansonsten wäre es beispielsweise möglich, wenn es eine Relation "Schwarzwald" gäbe die alle Objekte des Schwarzwaldes enthält, dass der Download eines einzigen Gebäudes oder einer Straße aus diesem Gebiet das herunterladen aller Objekte des Schwarzwaldes bewirken würde. Die Auswirkungen sind in Abbildung 3.7 sichtbar. Die Bahnstrecken des KVV sind zu Relations zusammengefasst worden. Durch das manuelle Nachladen aller in den Relationen enthaltenen Objekten erstreckt sich das Gebiet nun nicht mehr nur über die Karlsruher Innenstadt sondern über den gesamten von Linien des KVV bedienten Nordbadischen Raum.

¹⁵ Geofabrik <http://www.geofabrik.de>

¹⁶ OpenStreetMap API-Server: <http://api.openstreetmap.org/>

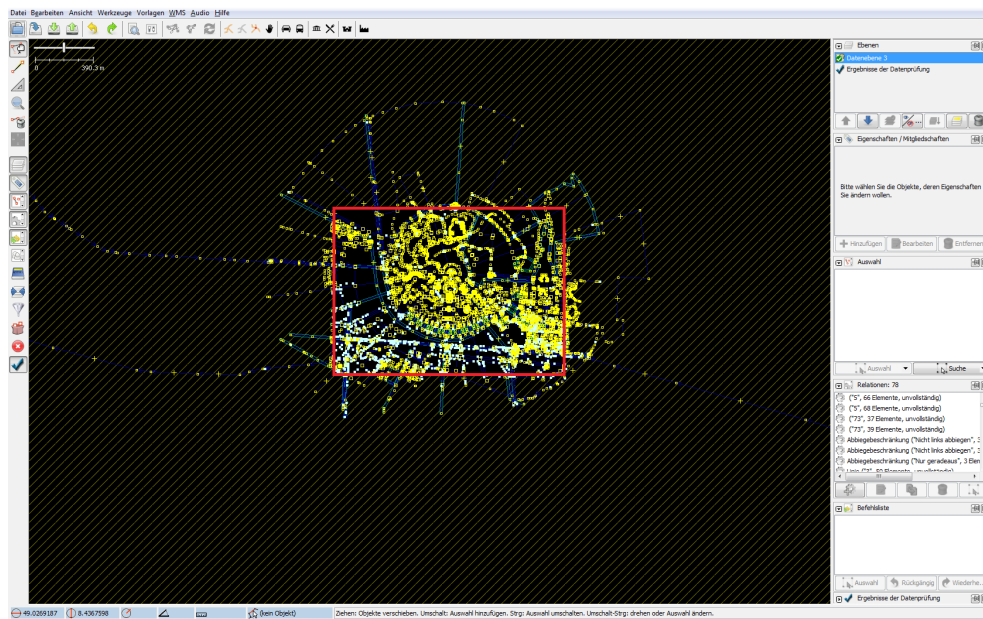


Abbildung 3.6: OSM Daten der Karlsruher Innenstadt. Der rote Rahmen zeigt den ausgewählten Bereich an.

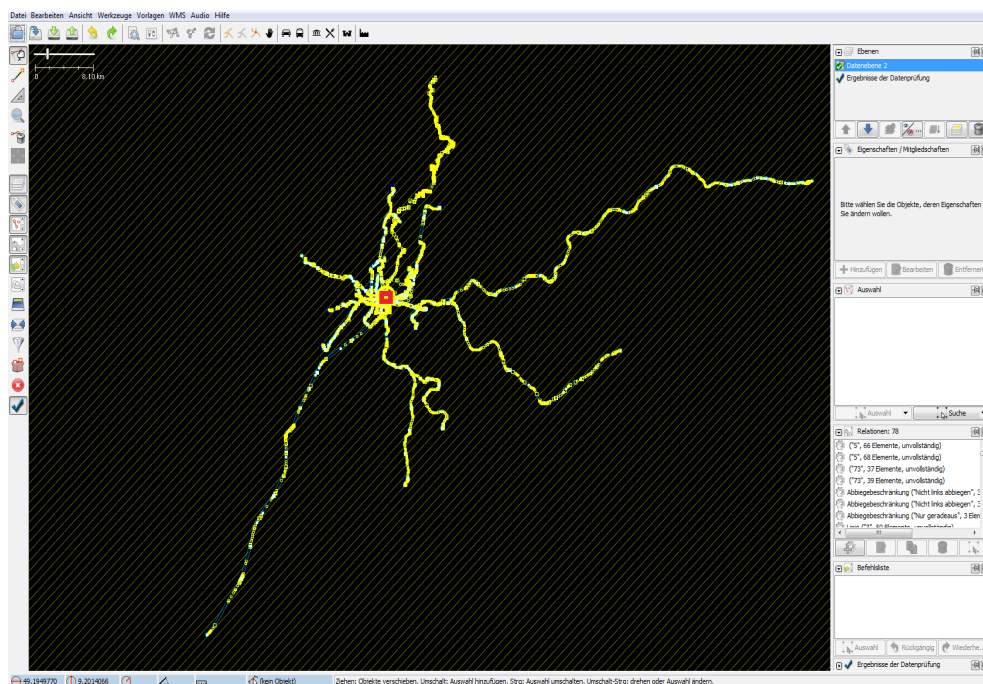


Abbildung 3.7: Der gleiche Bereich aber mit allen von den enthaltenen Relationen benutzten Nodes und Ways.

Mithilfe der API können Gebiete heruntergeladen werden die eine maximale Ausdehnung von 0,25 Quadratgrad haben (also Breite der BBox * Länge der BBox < 0,25 Grad²). Das entspricht in Deutschland einer Fläche von ungefähr 50 * 50 km². Außerdem darf der Ausschnitt nicht mehr als 50.000 Nodes enthalten. Größere Gebiete oder Gebiete mit einer hohen Dichte an Nodes müssen somit also stückchenweise heruntergeladen und zusammengesetzt werden.

Die API bietet auch eine Funktion um nur Objekte zu liefern die bestimmte Key-Value Paare besitzen. Allerdings lässt sich diese Abfrage nicht mit der BBox Abfrage kombinieren und kann daher nicht sinnvoll genutzt werden.

Wenn nicht die OSM Daten sondern die zugrundeliegenden GPS Tracks von Belang sind lautet die entsprechende HTTP GET Anweisung folgendermaßen:

```
GET /api/0.6/trackpoints?bbox=left,bottom,right,top&page=page
```

Die BBox wird mit denselben Parametern definiert wie auch schon bei der Kartenabfrage. Allerdings ist hier der Zusatz des zusätzlichen Parameters page zur Angabe der Seitenzahl notwendig, da maximal 5000 Trackpoints auf einer XML Seite ausgegeben werden können. Weitere Punkte im Gebiet können über das Erhöhen der Seitenzahl angesprochen werden. Wenn keine weiteren Punkte vorhanden sind wird eine leere Ergebnismenge zurückgeliefert.

Die API bietet darüber hinaus noch einige weitere Funktionen die aber für diese Arbeit nicht weiter von Belang sind. Eine komplette Übersicht ist im capabilities Dokument unter der folgenden Adresse zu finden:

```
http://api.openstreetmap.org/api/0.6/capabilities
```

Auch der JOSM Editor benutzt die API um die für eine Bearbeitung eines Gebietes benötigten Daten herunterzuladen. Allerdings stellt er dafür ähnlich wie das in Kapitel 3.1 beschriebene Programm EasyMap eine Karte zur Verfügung über die an die gewünschte Stelle navigiert werden kann. Dort kann dann durch eine BBox der gewünschte Ausschnitt weiter spezifiziert werden. Auch bietet JOSM die Möglichkeit Lesezeichen für bestimmte Regionen zu erstellen und abzulegen. So können große Gebiete sehr einfach in kleinere Teile zerlegt werden die nicht mehr unter die Größenbeschränkung fallen und somit heruntergeladen werden können. Wenn ein Gebiet nur über seinen Permalink bekannt ist kann JOSM die Parameter automatisch herauslesen und das entsprechende Gebiet herunterladen. Auch eine Namenssuche um etwa eine bestimmte Stadt zu finden ist möglich. Dabei ist jedoch zu beachten das die Suche in allen verfügbaren key-value Paaren nach korrespondierenden Zeichenketten sucht. So werden bei der Suche nach "Karlsruhe" neben Karlsruhe selbst und einigen Objekten in Karlsruhe wie etwa der Karlsruher Messe auch die Straßen "Karlsruhe Pl" in Missouri und "Karlsruhe Ave" in Kentucky gefunden (siehe Abbildung 3.8). Wenn gewünscht kann JOSM die Daten noch mit vorhandenen GPS Tracks, Luftbildern oder anderen .osm Daten kombinieren und auch als eine, entsprechend große, Datei speichern.

Name	Typ
	Stadt
Karlsruhe	village
Karlsruhe Pl	Straße
Karlsruhe Ave	Straße
Karlsruhe Cross Hwy	Straße
Flughafen Karlsruhe/Baden-Baden [FKB] [EDSB]	aerodrome
Flughafen Karlsruhe/Baden-Baden Terminal	terminalbuilding
Messe Karlsruhe-Forchheim	commercial area
Railway Karlsruhe-Rastatt	Eisenbahn
Railway Karlsruhe-Rastatt	Eisenbahn
Finanzamt Karlsruhe-Durlach	public buildingbuilding
Hauptzollamt Karlsruhe	public building
Bahnsteig Richtung Karlsruhe	steps
Hauptbahnhof (Karlsruhe)	car sharing
Karlsruhe Hauptbahnhof	station
Polizeiposten Karlsruhe-Oberreut	police
Polizeiposten Karlsruhe-Rüppurr	police
Volksbank Karlsruhe	bank
Fahrradstation am Hauptbahnhof Karlsruhe	bicycle parking
Karlsruhe-Mitte [45]	motorway junction
Messe Karlsruhe	Gebäude

Abbildung 3.8: Ergebnisse der Namenssuche von JOSM für "Karlsruhe".

OSM Extended API

Die OSM Extended API oder kurz XAPI ist eine modifizierte Version der eben vorgestellten API. Sie bietet erweiterte Such- und Abfragemechanismen hat dafür aber keinen Schreibzugriff auf die Datenbank, ist also eine read-only API. Ein weiterer Unterschied ist das die XAPI nicht direkt auf die zentrale OSM DB zugreift, sondern auf einen Mirror der Datenbank, der mit den minütlichen Diffs der zentralen DB aktualisiert wird (Näheres dazu in Kapitel 3.5). Dadurch sind die Daten im Regelfall jedoch nicht mehr als 10 Minuten älter als diejenigen die über die API abgefragt werden. Neben dem Standard XAPI-Server von OSM¹⁷ ist der Dienst auch über andere Server ansprechbar wie beispielsweise <http://www.informationfreeway.org/api/0.6/>.

Da die XAPI auf der Standard API aufbaut funktionieren auch alle im vorherigen Kapitel vorgestellten Abfragemechanismen entsprechend. Allerdings ist die Größe des möglichen Suchgebietes auf 100 Quadratgrad erhöht worden. Neu ist auch die Möglichkeit die BBox Abfrage mit der Abfrage nach bestimmten Schlüsseln zu kombinieren und diese nun sinnvoll einsetzen zu können um Objekte in abgegrenzten Gebieten zu finden. Durch die Kombination beider Abfragen ist es sogar möglich die Größenbeschränkung von 100 Quadratgrad aufzuheben und eine weltweite Suche zuzulassen. Dies ist sogar die Standardeinstellung die bei nicht näher spezifizierter BBox genutzt wird. Jedoch ist bisher nur eine Kombination aus einem BBox und einem Schlüsselprädikat

¹⁷ OSM XAPI-Server: <http://xapi.openstreetmap.org/api/0.6/>

möglich. Aber auch damit lassen sich bereits sinnvolle Suchanfragen stellen. Beispielsweise suchen die beiden folgenden Abfragen alle Bänke im Bereich des Berliner Tiergartens oder alle Krankenhäuser im Vereinigten Königreich:

- `http://xapi.openstreetmap.org/api/0.6/*[amenity=bench][bbox=13.33,52.51,13.38,52.52]`
- `http://www.informationfreeway.org/api/0.6/node[amenity=hospital][bbox=-6,50,2,61]`

Dabei sucht * beliebige Elemente die mit dem entsprechenden Tag versehen sind. Stattdessen könne auch nur bestimmte Objektarten gesucht werden indem * durch Node, Way oder Relation ersetzt wird. Werden Ways oder Relations gefunden werden im zurückgegebenen XML Dokument auch alle von diesen referenzierten Nodes oder Ways mitgeliefert. Es ist auch möglich einen Vereinigungsoperator bei der Schlüsselsuche zu verwenden. Die beiden folgenden Abfragen finden so einmal alle als Autobahn, Schnell oder Bundesstraße (entspricht den Attributen highway, trunk und primary) getaggten Ways oder alle Nodes die Golfkurs entweder als Einrichtungs- (amenity) oder als Freizeit (leisure) Schlüssel haben

- `.../way[highway=motorway|motorway_link|trunk|primary]`
- `.../node[amenity|leisure=golf_course]`

Da durch die aufgehobene Gebietsbeschränkung die entstehenden Datenmengen sehr groß werden, sollten diese Anfragen nicht mit einem Webbrowser durchgeführt werden da diese mit diesen Datengrößen oft nicht umzugehen vermögen. Stattdessen wird die Verwendung von spezialisierten Zusatzprogrammen wie curl¹⁸ oder wget¹⁹ empfohlen.

Das Planet File

Einmal wöchentlich wird auch das sogenannte Planet File (planet.osm) angeboten. Dabei handelt es sich um einen Schnappschuss der gesamten OSM Datenbank mit allen Nodes, Ways und Relations. Die Größe des gesamten Datenmaterials betrug im September 2010 über 171 GB, komprimiert durch das freie Komprimierungsprogramm bzip2 auf 11GB. Zur Aktualisierung dieses Abzuges werden regelmäßig Aktualisierungsdateien, sogenannte Diffs angeboten. Dabei handelt es sich um Änderungsdatensätze welche die seit der letzten Aktualisierung geänderten Objekten beinhalten und die als tägliche, stündliche und minütliche Datensätze vorhanden sind. Mehr dazu in Kapitel 3.5. Einige Mirror Seiten bieten neben dem Planet file auch Auszüge für bestimmte Länder oder Regionen an. So bietet Geofabrik auf ihrer Download Seite²⁰ neben Auszügen für Kontinente, Länder auch Auszüge für deutsche, englische und französische Bundesländer und in Bayern und Nordrhein-Westfalen sogar für einzelne Städte und Regionen an

18 Curl: <http://curl.haxx.se/>

19 Wget: <http://www.gnu.org/software/wget/>

20 Geofabrik Download Seite: <http://download.geofabrik.de/osm/>

3.3 Thematische Datenauswahl

Neben der Räumlichen Auswahl die wie in Kapitel 3.2 über die API beziehungsweise über die XAPI durchgeführt wird ist meist noch eine thematische Auswahl vonnöten die über die Auswahl eines einzelnen key-value Paares wie es bei der XAPI möglich ist hinausgeht. Auch dafür wurden in der OSM Gemeinde verschiedene Lösungen entwickelt. Aber alle leiden unter der Tatsache das dafür eine komplette .osm Datei des Gebietes vorliegen muss und somit, je nachdem wie groß das Gebiet ist und wie stark die thematische Auswahl eingrenzt werden soll, viele unnötige Daten heruntergeladen werden müssen. Zwei dieser Programme werden im folgenden genauer erklärt.

excerpt_tags

Eine der Möglichkeiten eine vorliegende .sm Datei thematisch auszudünnen ist das Perl Script `excerpt_tags`²¹, das auch im SM Repository vorliegt. Zur Ausführung des Scripts muss ein Perl Interpreter, wie beispielsweise Active Perl²², installiert sein. Zudem werden zusätzliche Module und Bibliotheken benötigt, die alle aus dem SM Repository²³ heruntergeladen werden können. Die Thematische Auswahl der Daten wird dann im Quellcode selbst vorgenommen. Dazu wird das Script mit einem Texteditor geöffnet. Im User Configuration Block am Anfang des Codes findet man dann drei Variablen in denen dann die entsprechenden key-value Paare für Nodes, Ways und Relations eingetragen werden. Um so beispielsweise nur Eisenbahnstrecken und Autobahnen herauszufiltern muss folgendes eingetragen werden.

```
my @way_sel_tags = (
    ['railway', undef],
    ['highway', 'motorway'],
);
```

Wenn das Script angepasst ist kann es über die Kommandozeile mit folgendem Befehl aufgerufen werden:

```
perl planetosm-excerpt-tags.pl [osm-datei] > [ausgabe-datei]
```

Die entstehende Ausgabedatei ist in den meisten Fällen jedoch unsortiert und kann so nicht mit jedem Programm problemlos weiterverarbeitet werden. Hier kann das Programm Osmosis Abhilfe schaffen.

Osmosis

Osmosis ist eines der wichtigsten Werkzeuge der OSM Gemeinde. Diese von Brett Henderson in Java geschriebene Kommandozeilenprogramm beherrscht neben der Möglichkeit thematische Auszüge aus einer .osm zu erstellen noch viele weitere Funktionen die für komplexeres Arbeiten mit OSM Daten nützlich sind. Dazu gehören:

- Lese- und Schreibmöglichkeiten für .osm Dateien sowie für MySQL- und PostgreSQL Datenbanken

²¹ `excerpt_tag`: <http://svn.openstreetmap.org/applications/utils/osm-extract/planetosm-excerpt-tags.pl>

²² Active Perl: <http://www.activestate.com/activeperl>

²³ Perl Module: http://svn.openstreetmap.org/applications/utils/perl_lib/

- Ermittlung von Änderungen zwischen 2 .osm Dateien oder 2 Zeitpunkten einer Datenbank und erzeugen der zugehörigen Diff Dateien (siehe Kapitel 3.5)
- Einpflegen von Änderungsdatensätzen in Datenbanken oder bestehende .osm Dateien.
- Ausschneiden einer BBox oder eines Polygonumrisses aus einem bestehenden Datensatz
- Filterung nach bestimmten key-value Paaren von Nodes und Ways
- Sortierung von .osm Dateien

Osmodis befindet sich immer noch in der Weiterentwicklung und kann durch seinen modularen Aufbau einfach um weitere Funktionen ergänzt werden. Daher ist die obige Aufzählung von möglichen Funktionen nicht abschließend. Für die Ausführung von Osmodis wird die Java Laufzeitumgebung JRE 1.6 benötigt. Für Systeme die diese Version nicht installieren können ist auch eine eingeschränkte Version die zu JRE 1.5 kompatibel ist verfügbar.

Osmodis arbeitet grundsätzlich mit einem so genannten "Pipeline-Modus". Das bedeutet das Eingabedaten nicht zuerst komplett eingelesen, anschließend verarbeitet und am Ende wieder ausgegeben werden. Sondern die Daten werden Stück für Stück abgearbeitet. Dies ermöglicht den problemlosen Einsatz auch bei beliebig großen Dateien, wie etwa dem Planet File. Ein Osmodis Aufruf besteht daher immer aus mindestens 2 Parametern, nämlich einer Datenquelle für die Eingabe und einem Ziel für die Ausgabe. Ein solcher Aufruf, der zugegebenermaßen keine sehr sinnvolle Arbeit verrichtet da er lediglich den Inhalt der Eingabedatei in die Zieldatei schreibt, sieht folgendermaßen aus:

```
osmodis --read-xml file=<quelldatei> --write-xml  
file=<zieldatei>
```

Jedes mögliche Modul wird mit einem solchen, von zwei Bindestrichen eingeleiteten, Kommandozeilenoption angesprochen. Dabei benötigen viele Module mehrere Parameter die jeweils mit der Form Parameter=Wert hinter dem jeweiligen Modul angegeben werden müssen. Eine Erläuterung aller in Osmodis nutzbaren Module findet sich im [OpenStreetMapWiki], in [Ramm2008] ist auch eine gute Übersicht der zum damaligen Zeitpunkt der Drucklegung aktuellen Version vorhanden. Diese ist mittlerweile zwar nicht mehr auf dem neuesten Stand der Entwicklung, für die meisten Aufgaben ist sie aber durchaus noch ausreichend da der Großteil der Standardmodule nicht verändert wurde.

Für die Ausgabe unterstützt Osmodis auch die Kompressionsverfahren gzip und bzip2 die automatisch benutzt werden wenn die Ausgabedatei die entsprechende Endung aufweist. Dabei ist die gzip Variante deutlich schneller als die bzip2 Version da die Java Implementierung der bzip2 Algorithmen nicht mit den nativen Code von gzip der auch unter Java nutzbar ist mithalten kann. Für die bzip2 Kompression größerer Daten wird daher die Verwendung eines externen Packprogramms empfohlen.

3.4 Bearbeitung

Die Bearbeitung von OSM Daten kann grundsätzlich auf 2 verschiedenen Wegen erfolgen. Zum einen online mit dem in die OSM Seite integrierten Flash Editor Potlach der direkt die dahinter liegende OSM DB modifiziert oder man lädt sich den benötigten Ausschnitt über eine der in Kapitel 3.2 angesprochenen Methoden herunter und bearbeitet diesen mit JOSM oder einem GIS Programm das über entsprechende Bearbeitungswerkzeuge verfügt. Ein GIS Programm kann dabei entweder

direkt mit .osm Daten arbeiten, zu diesen gehört beispielsweise QuantumGIS (QGIS)²⁴, oder man benutzt Osmosis oder eines der vielen weiteren verfügbaren Tools wie etwa das in C und C++ geschriebene Hilfsprogramm osm2pgsql²⁵ um OSM Daten in eine Datenbank zu schreiben und bindet diese dann in ein GIS seiner Wahl ein. Mit QGIS ist auch diese Variante möglich. Wenn die gemachten Änderungen am Ende in die OSM Datenbank zurück gespielt werden sollen um sie der OSM Gemeinde zugänglich zu machen empfiehlt sich JOSM oder ein GIS mit implementierten .osm Bearbeitungswerkzeugen da der Weg von OSM über eine eigene DB zur Bearbeitung in ein GIS und wieder zurück deutlich komplexer ist. Falls Änderungen jedoch nur für das eigene Projekt gemacht werden sollen hängt die Auswahl des Verwendeten Programms lediglich von den eigenen Vorlieben ab. Die Anzahl und Möglichkeiten der Bearbeitungswerkzeuge hängt dabei naturgemäß von der Auswahl des Programmes ab und muss den erforderlichen Bedürfnissen entsprechend bei der Auswahl des Programmes mitberücksichtigt werden.

So kann QGIS wenn die originären OSM Bearbeitungswerkzeuge benutzt werden die Objekte nur mit dem beim Einlesen der Daten gewählten Darstellungsstil visualisieren. Eine Änderung der Darstellung ist nicht möglich da alle Tags intern als ein einziges Attribut gespeichert werden und so mit eine Unterscheidung zwischen verschiedenen Tags zu Darstellungszwecken nicht mehr möglich ist. Jeder möglichen Kombination von Tags müsste so eine individuelle Darstellung zugewiesen werden. Und schon bei einem Ausschnitt von der Größe des KIT Campus Süd kämen dabei über 1000 verschiedene Stile zustande. Dafür erfolgt die Bearbeitung der Tags auf diese Weise mit nur wenigen Mausklicks, während bei der Variante über eine PostgreSQL Datenbank zunächst das entsprechende Objekt aus der Datenbank gesucht werden muss um anschließend jeden Tag einzeln zu ändern. Auch können auf diese Weise nur Nodes und Ways bearbeitet werden, Relations gehen beim Import in die Datenbank verloren.

3.5 Aktualisierung

Zusätzlich zum .osm Format für die Rohdaten gibt es noch ein weiteres XML basiertes Datenformat namens OSMChange (.osc) das Änderungen an einem .osm Datensatz beschreibt. Dazu kann ein .osc File beliebig viele <create>, <modify> und <delete> Blöcke enthalten in denen die jeweils zu erstellenden, bearbeitenden und zu löschenden Objekte beschrieben sind die seit der vorherigen Version verändert wurden.

Diese Änderungsdateien werden von OSM als tägliche, stündliche und sogar minütliche Differenzdaten, die sogenannten Diffs, bereitgestellt die jeweils die Änderungen zum vorherigen Zeitpunkt enthalten. Dadurch ist es einfach einen bereits bestehenden OSM Datenbestand aktuell zu halten. Für das planet.osm hat ein tägliches update beispielsweise weniger 0,5% der Größe der gesamten Datei, die stündlichen meist nur noch 0,1%. Durch die geringe Größe reduziert sich die Download- sowie die Einpflegezeit in eine Datenbank im Vergleich zu einem kompletten Datenbankabzug um ein Vielfaches.

Da die Zeitpunkte der Erstellung des Planet Files und der Änderungsdateien nicht synchron sind empfiehlt es sich beim Aufsetzen einer Datenbank auf jeden Fall noch die Änderungsdatei des vorherigen Zeitraumes mit einzupflegen. Da durch eine Änderungsdatei nur diejenigen Änderungen übernommen werden, die im Datenmaterial noch nicht bekannt sind richtet selbst das Einspielen zu alter Änderungsdateien keinen Schaden an.

24 QGIS: <http://qgis.org/>

25 osm2pgsql: <http://svn.openstreetmap.org/applications/utils/export/osm2pgsql/>

Solche .osc Dateien können auch mit Osmosis für beliebige Zeiträume selbst erstellt werden, wenn die entsprechenden OSM Daten für die vorherige und nachherige Version vorliegen. Dies kann sinnvoll sein da OSM die Diffs nicht für unbegrenzte Zeit zur Verfügung stellt. Meistens empfiehlt es sich jedoch die von OSM bereitgestellten Diffs zu nutzen und diese regelmäßig in das bestehende Datenmaterial einzupflegen. Dafür bietet Osmosis die Möglichkeit über einen "cron"-Job, also einen regelmäßig ausgeführten Befehl, die Diffs herunterzuladen und in den vorhandenen Datenbestand einzupflegen.

Für kleine Gebiete die schon zu einem Hohen Grad erfasst worden sind kann es auch sinnvoll sein auf solche Updates zu verzichten da hier die Diffs oftmals keine Änderungen enthalten und ein erneutes Herunterladen des gesamten Gebietes in größeren Abständen nur wenig langsamer ist dafür keinerlei zusätzlichen Konfigurations- und Wartungsaufwand nach sich zieht.

Wer sich bei kleineren Projekte dabei nicht nur auf die OSM Mapper verlassen will kann natürlich jederzeit selbst zu einem GPS-fähigen Gerät greifen und dafür sorgen das das betreffende Gebiet in der OSM Datenbank auf dem aktuellsten Stand ist indem die erfassten Änderungen sowohl in den eigenen als auch den OSM Datenbestand eingepflegt werden.

4 Fallbeispiel Campusplan

In [Veit2010] wurden bereits dargelegt, welche Informationen in einem Campusplan benötigt werden. Dabei wurde besonders deutlich, dass für verschiedene Nutzergruppen auch verschiedene Informationen und Informationsgrade benötigt werden. Konzis zusammengefasst ergeben sich daraus folgende Informationen die in einem Campusplan darzustellen sind.

- Gebäude
 - Gebäudenummern / Funktionsbezeichnungen
 - Stattfindende Veranstaltungen
 - Interner Aufbau
 - Institute
 - Name
 - Mitarbeiter
 - Hörsäle
 - Name
 - Stattfindende Vorlesungen
 - Foto / 3D-Ansicht
- Straßen
 - Straßennamen
 - Fußgängerzonen
 - Einbahnstraßen
- Wegmarken
 - Statuen
 - Plätze
 - Bäume
- Infrastruktur
 - Parkmöglichkeiten
 - Nahverkehr
 - Öffnungszeiten

Ein Großteil dieser benötigten Informationen, genauer gesagt alle mit direktem Geometriebezug, wie Wegmarken, Straßen oder Gebäude, sind für den KIT Campus bereits in OSM erfasst worden. Für solche Informationen ist auch eine Erfassung fehlender oder Aktualisierung bereits vorhandener Elemente nicht weiter schwierig, da hier jeder im Rahmen des OSM Projektes selbst Hand anlegen kann. Andere Informationen, etwa die zu Öffnungszeiten, Mitarbeitern oder Gebäudeansichten sind nicht Teil von OSM und müssen daher aus anderen Quellen kommen.

4.1 Beitrag von OpenStreetMap

Wie bereits erwähnt sind in OSM Informationen mit Geometriebezug, also die Basisdaten eines Plans welche für alle Benutzer Relevanz haben, verfügbar oder können mit nur geringem Aufwand dort hinterlegt werden. Zu diesen Basisdaten gehören unter anderem Straßenverlauf und Straßennamen sowie zusätzliche Informationen zu diesen (etwa Abbiegevorschriften, Fußgängerzonen oder Einbahnstraßen) um einen effektiven Routingservice damit betreiben zu können. Auch die vorhandenen oder geplanten Gebäude und ihre Bezeichnungen (Gebäudenummern) oder Funktionen (beispielsweise Mensa, Rechenzentrum oder Studienbüro) sowie sonstige POI wie Parkplätze, Statuen, Parks oder Plätze sind solche Basisdaten.

Daraus kann entweder eine eigene Basiskarte generiert werden oder die bereits bestehenden OSM Karten werden als Grundlage verwendet. Dabei ist aber zu beachten das dann keine Auswahl der darzustellenden Objekten getroffen werden kann. Außerdem ist man an den von OSM vorgegebenen Kartenstil gebunden. Beide Einschränkungen können mit den in den vorhergehenden Kapiteln beschriebenen Maßnahmen angepasst werden.

Die im Gebiet des KIT Campus Süd verwendeten Tags sind in Tabelle 4.1 aufgeführt. Die so getaggten Objekte können für eine Basiskarte verwendet werden. Falls die Basiskarte jedoch nicht selbst generiert wird sondern die von OSM vor gerenderten Kartenbilder benutzt werden, können einige der Objekte aufgrund fehlender Zeichenvorschriften nicht dargestellt werden. Tabelle 4.1 ist auch lediglich für den Zeitpunkt der Erstellung abschließend, da jederzeit neue Objekte erfasst und mit hier nicht aufgeführten Tags versehen werden können.

Die hier aufgeführten Tags wurden mit `excerpt_tags` aus einer die Karlsruher Stadtmitte umfassenden `.osm` Datei gefiltert. Anschließend mit `Osmosis` sortiert und mit `osm2pgsql` in eine PostgreSQL Datenbank eingepflegt die dann in QGIS eingebunden und dargestellt wurde (Abbildung 4.1). Dabei werden geschlossene Ways automatisch in Polygone umgewandelt und auf einen eigenen Layer ausgelagert. Der Layer `roads` wird dabei von `osm2pgsql` zusätzlich angelegt, um das Kartenrendern mit Mapnik, das ebenfalls auf eine PostgreSQL Datenbank zugreift, zu beschleunigen. Im vorliegenden Beispiel ist das aber ohne Bedeutung.

Für die Visualisierung muss anschließend noch ein aussagekräftiger Darstellungsstil erstellt werden. Für die Darstellung von einzelnen Nodes werden von QGIS dabei etliche Symbole mitgeliefert die bestimmten Tags zugeordnet werden können. Die Zuordnung in QGIS erfolgt dabei mit einem regelbasierten Filter in welchem einem key-value Paar ein Symbol zugeordnet wird. Für Linien und Flächen gilt dies entsprechend, nur das Anstelle eines Symboles die Farben und Stärken der Linien einem bestimmten Tag zugeordnet werden. Für Polygone muss neben den Eigenschaften der Umrandungslinien auch die Farbe und Transparenz der Füllung gewählt werden.

Tabelle 4.1: Für KIT Campus Süd verfügbare OSM-Tags.

Key	Value
natural	tree, scrub, water
amenity	bicycle_parking, emergency_phone, parking, auditorium, telephone, cafe, restaurant, biergarten, library, school, bench, car_sharing, fast_food, vending_machine, atm, recycling
highway	footway, residential, living_street, service, path, unclassified, pedestrian, bus_stop, steps, crossing, construction, cycleway
leisure	park, water_park
historic	memorial
building	yes, entrance
barrier	bollard, wall, lift_gate, gate, fence, entrance, cycle_barrier
landuse	grass, construction
shop	hairdresser
information	office
man_made	surveillance
type	multipolygon

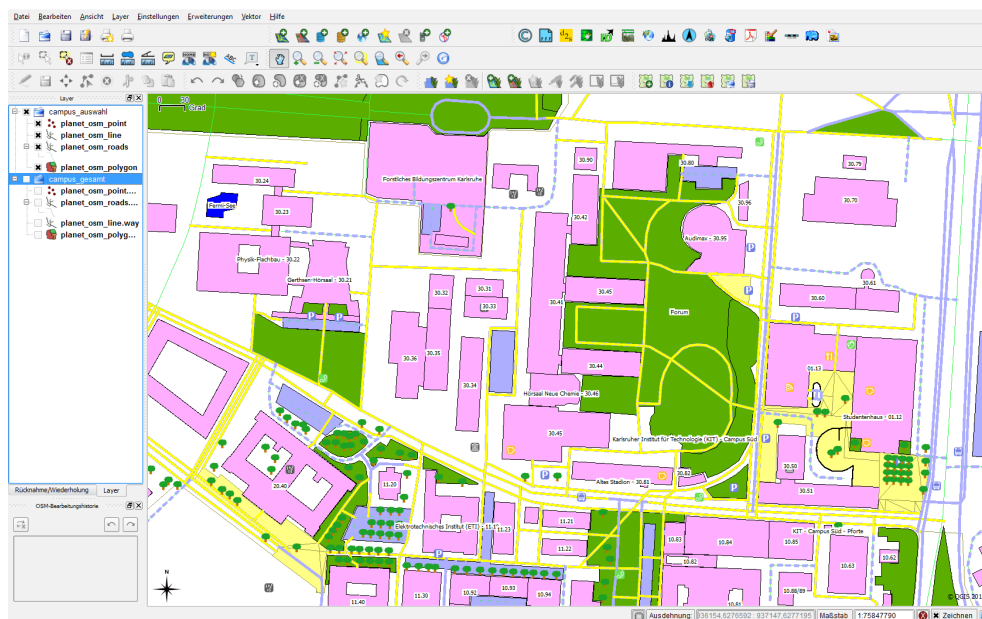


Abbildung 4.1: Darstellung der nach Tabelle 4.1 extrahierten Daten in QGIS.

4.2 Kartenoverlays

Die so entstandene Karte beinhaltet aber nur die im OSM Datenbestand vorhandene Objekte die, wenn entsprechende Zeichenvorschriften für die Objekte in den Kartenrenderern hinterlegt sind, auch schon in den von OSM generierten Karten dargestellt werden können. Je nach Grad der thematischen Auswahl kann es sogar vorkommen dass weniger Informationen dargestellt sind. Für einen Campusplan der alle in [Veit2010] spezifizierten Informationen enthalten soll werden weitergehende Informationen benötigt. Zum Beispiel die in einem Gebäude vorhandenen Hörsäle und die darin stattfindenden Veranstaltungen oder die Fahrplaninformationen des öffentlichen Nahverkehrs. Viele dieser zur Komplettierung benötigten Informationen sind bereits online verfügbar, jedoch zumeist über viele verschiedene Webseiten und Webdienste verteilt. Diese müssen für die Einbindung in den Plan in ein, für ein Overlay der Basiskarte benötigtes Format gebracht werden und als zusätzliche Datenquelle eingebunden werden. Dies kann beispielsweise geschehen indem im Hörsaalverzeichnis des KIT jeder Hörsaal mit Geodaten zum seinem Ort auf dem Campus versehen und diese Informationen durch einen WFS (WebFeatureService) online gestellt wird. Auch das Vorlesungsverzeichnis kann auf diese Weise mit den entsprechenden Hörsälen und damit auch mit den auf der Karte sichtbaren Gebäuden verknüpft werden. Solch ein WFS kann dann mittels OpenLayers einfach als zusätzlicher Datenlayer über die Basiskarte gelegt werden. Oder aber diese Informationen werden in einer eigenen Datenbank gespeichert die mit der Karte verknüpft wird, sodass durch Klicken auf die Karte auf die entsprechenden Informationen zugegriffen werden kann.

5 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es einen Überblick darüber zu geben wie aus “Crowd Sourced Cartography“ [Coast2009] wie es OSM Gründer Steve Coast plakativ ausdrückte eine verwertbare Datengrundlage für eigene Karten und Projekte geschaffen werden kann. Beispielhaft für ein solches Projekt ist [Neis2008] zu nennen.

Dafür wurde verschiedene Wege vorgestellt auf denen die OSM Daten akquiriert werden können. Die Direktzugriffsvariante ist dabei auch für technisch weniger versierte ohne größere Probleme nutzbar. Jedoch ist sie nur für wenige Projekte geeignet da Möglichkeiten um die Daten den projektbezogenen Erfordernissen anzupassen nicht gegeben sind. Die Alternative, der Download eines geographisch und thematisch eingegrenzten Ausschnittes des Datenmaterials bietet diese Möglichkeiten. Dafür muss neben der Auswahl und Bearbeitung der Daten auch für die Aktualität selbst gesorgt werden.

Neben Datenzugriff, der geometrischen und thematischen Auswahl von Daten sowie der eventuell notwendigen Nachbearbeitung und regelmäßigen Aktualisierung die im Rahmen dieser Arbeit ausführlich beschrieben wurden benötigt eine Karte natürlich noch eine ansprechende Visualisierung. Hierbei wurde in der Arbeit lediglich auf die Visualisierung durch ein GIS näher eingegangen. Um die in Kapitel 4.1 beispielhaft erstellte Karten des KIT Campus aber nicht nur lokal mit QGIS oder einem ähnlichen Programm betrachten zu können, sondern in eine online verfügbare “Slippy Map“ einbinden zu können muss die Karte gekachelt vorliegen, also aus vielen einzelnen mosaikartigen Rasterbildern bestehen die dann mit OpenLayers in einen webbasierten Plan eingebunden werden können. Erst dann können auch zusätzlichen Daten in Form von Overlays mit in die Karte integriert werden. Die hierfür notwendigen Programme und Vorgehensweisen ausführlich zu untersuchen und vorzustellen würde allerdings den Rahmen der vorliegenden Arbeit sprengen und benötigt daher eine gesonderte Betrachtung.

Abschließend kann mit dieser Arbeit bestätigt werden das die vom OpenStreetMap-Projekt zusammengetragenen Geodaten sich mittlerweile als ernsthafte Alternative zu den etablierten Anbietern von Karten und Geodaten präsentieren. Die Vorteile liegen vor allem in den “frei“-verfügbaren Geodaten aber auch darin das jeder leicht Ergänzungen und Verbesserungen am vorhandenen Datenmaterial vornehmen kann und so die Aktualität der Daten denen anderer Anbieter oftmals voraus ist. Allerdings unterliegt die Abdeckung weltweit starken Schwankungen was die Vollständigkeit der Daten betrifft. Für eine Nutzung von OSM Daten muss daher zunächst das vorhandene Material gesichtet werden um abwägen zu können ob eine Nutzung sinnvoll erscheint. Ist dies der Fall steht einer Nutzung durch die in dieser Arbeit beschriebenen Möglichkeiten und den vorgestellten Programme nichts mehr im Weg.

Literaturverzeichnis

Coast, Steve(2009): Interview: How maps will be free and satnavs much cheaper, <http://www.pocket-lint.com/news/26612/osm-openstreetmap-interview-steve-coast> (18.12.2010)

Jansen, Marc; Adams, Till (2010): OpenLayers : Webentwicklung mit dynamischen Karten und Geodaten, Open Source Press, München

Neis, Pascal (2008): Location Based Services mit OpenStreetMap Daten, Masterarbeit, Fachhochschule Mainz Fachbereich I

O'Reilly, Tim(2005): What is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software, <http://oreilly.com/web2/archive/what-is-web-20.html> (18.12.2010)

OpenStreetMap: <http://www.openstreetmap.org/> (18.12.2010)

OpenStreetMapWiki: http://wiki.openstreetmap.org/wiki/Main_Page (18.12.2010)

Ramm, Frederik; Topf, Jochen (2008): OpenStreetMap: die freie Weltkarte nutzen und mitgestalten, Lehmanns Media, Berlin

Veit, Carla (2010): Konzept für die Neugestaltung des webbasierten Campusplan des KIT, Studienarbeit, Institut für Photogrammetrie und Fernerkundung (IPF), Karlsruher Institut für Technologie (KIT)

Abkürzungsverzeichnis

API	Application Programming Interface
BBox	Bounding Box
DB	Datenbank
GIS	GeoInformationSystem
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identifikator
JRE	Java Runtime Environment
KML	Keyhole Markup Language
KVV	Karlsruher Verkehrsverbund
OSC	OSMChance
OSM	OpenStreetMap
POI	Point of Interest
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
WFS	Web Feature Service
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation