

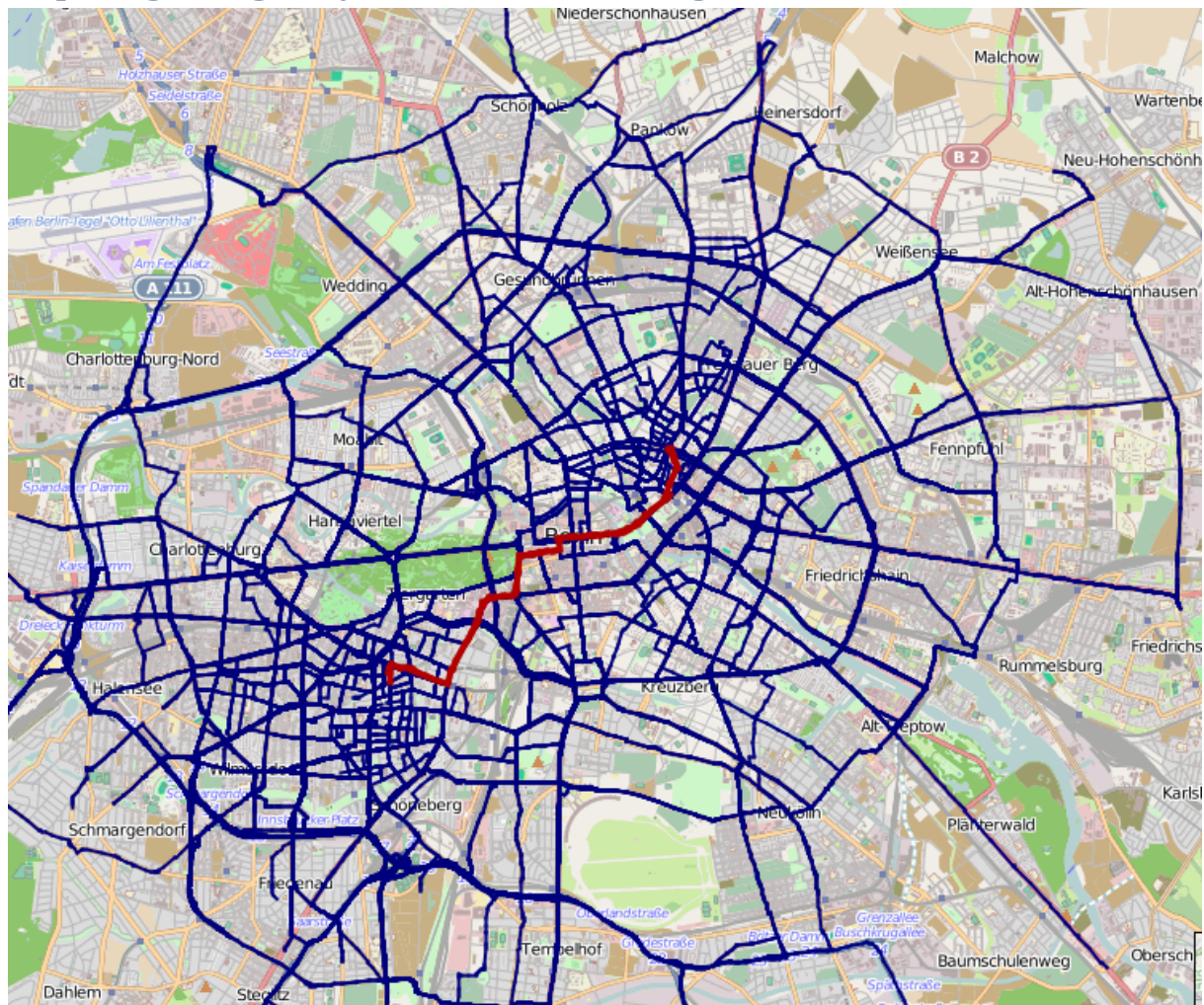


This document is licensed under the Creative Commons **Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)** License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>; or, (b) send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.”

Copyright (C) 2010 Fraunhofer Institute for Open Communication Systems (FOKUS)  
Fraunhofer FOKUS  
Kaiserin-Augusta-Allee 31  
10589 Berlin  
Tel: +49 30 3463-7000  
[info@fokus.fraunhofer.de](mailto:info@fokus.fraunhofer.de)

Das Kartenmaterial ist dem OpenStreetMap-Projekt entnommen und steht unter der Lizenz Creative Commons Attribution Share Alike-Lizenz 2.0.

## Mapsforge - Highway Hierarchies Routing



Im Folgenden eine kurze Anleitung wie Kartenmaterial vorberechnet und im OpenRideServer verwendet werden kann. Für die Vorbereitung wurde ein separates Projekt im Subversion Repository angelegt.

## Vorverarbeitung der OpenStreetMap Daten

Zur Vorverarbeitung müssen drei Teilschritte gemacht werden, Download des Kartenmaterials, Extraktion des Routinggraphen und die Berechnung der Hierarchies. Das Projekt dazu liegt hier: <https://openride.svn.sourceforge.net/svnroot/openride/trunk/src/OpenRideRouting>

Nach Anpassen der Konfiguration kann die Vorberechnung mit dem Ant-Task ‚produce‘ des Ant-Scripts ‚build-routing.xml‘ ausgeführt werden.

### Download des Kartenmaterials

Bei [www.openstreetmap.org](http://www.openstreetmap.org) kann man das komplette planet.xml herunterladen.

Zusätzlich gibt es mehrere Anbieter welche Ausschnitte einzelner Region anbieten. Bisher wurden die Daten von Geofabrik verwendet, welche hier ‚<http://download.geofabrik.de/osm/>‘ zum Download bereitstehen.

### Extraktion des Routinggraphen

Basierend auf dem Openstreetmap xml wird ein Routinggraph berechnet. In diesem Schritt werden anhand einer Whitelist nicht benötigte Straßentypen herausgefiltert. Eine Übersicht über alle Straßentypen findet sich hier ‚<http://wiki.openstreetmap.org/wiki/Key:highway>‘. Die Extraktion erfolgt mit dem Tool osm2rg welches im mapsforge-routing.jar enthalten ist. Osm2rg lässt sich mit Hilfe einer Properties Datei konfigurieren. Ein Beispiel findet sich unter ‚res/osm2rg.properties‘. Innerhalb dieser Properties Datei wird auch die Whitelist als Komma separierte Liste von ‚highway-tags‘ spezifiziert. Hier kann die Quelle (Osm-Datei) und die Senke (Datenbank) sowie die Whitelist konfiguriert werden. Vor der Extraktion muss das entsprechende Datenbankschema angelegt werden. Die Definition findet sich unter ‚res/osm2rgCreateTables.sql‘. Zum extrahieren des Graphen kann das Ant-Target ‚extract-rg‘ verwendet werden.

### Vorberechnung der Highway Hierarchies

Basierend auf dem Routinggraph wird jetzt eine Hierarchie von Graphen erzeugt und sowohl in einer Datenbank als auch in einem Flat Binary File gespeichert. Als Eingabe dient der bereits extrahierte Routinggraph welcher sich in der Datenbank befindet. Die Konfiguration erfolgt über eine Properties Datei, z.B. ‚res/hhPreprocessing.properties‘. Hier muss Eingabe Datenbank, Ausgabe Datenbank und Ausgabe Datei, Kantengewichtung sowie Tuningparameter des Algorithmus festgelegt werden. Zum Gewichten der Kanten wird in einer separaten Konfigurationsdatei eine Abbildung von Straßentyp auf Durchschnittsgeschwindigkeit definiert, z.B. ‚res/highwayLevel2AverageSpeed.txt. Die Tuning Parameter des Algorithmus sind bereits sinnvoll gewählt und können bei Bedarf hier ‚<http://algo2.iti.kit.edu/1068.php>‘ nachgelesen werden. Für die Zieldatenbank muss ein Schema angelegt werden ‚res/hhCreateTables.sql‘. Zum extrahieren kann das Ant-Target ‚compute-hh‘ verwendet werden.

## Verwendung der Mapsforge-routing Library

Es werden weitere Bibliotheken benötigt:

- mapsforge-routeing-0.1.0.jar
- postgresql-8.4-701.jdbc4.jar
- trove-2.1.0.jar
- trove-3.0.0.a3.jar

Schnittstelle – Routing:

- `org.mapsforge.server.routing.IRouter`
- `org.mapsforge.server.routing.IEdge`
- `org.mapsforge.server.routing.IVertex`

Erstellen eines IRouter Objekts (als Parameter ein Stream zur Routinggraph Datei)

- `org.mapsforge.server.routing.highwayHierarchies.RouterImpl.deserialize(InputStream is)`

### Einbindung in OpenRideServer-ejb

Die Mapsforge Routenplanung wird mit der Klasse `de.fhg.fokus.openride.routing.RouterBean` über das gleiche Interface bereitgestellt wie zuvor mit `pgRouting`. Dieses sollte aber bei Gelegenheit überarbeitet werden. Es gibt zwei Methoden zur Routenplanung wovon eine der beiden alle Koordinaten der Route liefert, die andere (welche für das Routenmatching verwendet wird) liefert weniger und dafür interpolierte Koordinaten. Dies hat zur Folge, dass jede Routenberechnung 2x erfolgt, 1x um die Route auf einer Karte Anzeigen zu können und ein weiteres Mal um die interpolierten Routenpunkte zu erhalten. Die Interpolation sollte deshalb aus dem `RouterBean` herausgezogen werden.

### Update des Kartenmaterial

Neues Kartenmaterial muss, wie in dem Abschnitt zur Vorverarbeitung beschrieben, vorberechnet werden. Das Ergebnis davon ist eine `.hh` Datei, welche von dem Router geladen wird. Um den Applicationserver nicht neu starten zu müssen, überwacht die Klasse `de.fhg.fokus.openride.routing.RouterWrapper` die Datei `routingGraph.hh` im Wurzelpfad der jeweiligen Domäne und lädt diese bei Änderungen neu. Somit reicht es aus, diese Datei einfach durch eine neue zu ersetzen, das Laden erfolgt dann innerhalb eines Hartkodierten Intervalls z.Zt. 30s.