Comparison of Open Source routing services with OpenStreetMap Data for blind pedestrians

PgRouting, OpenTripPlanner and OpenSourceRoutingMaschine

Markus Dornhofer
FH JOANNEUM
Energy and Transportation
Kapfenberg - Austria
markus.dornhofer
@fh-joanneum.at

Werner Bischof
FH JOANNEUM
Energy and Transportation
Kapfenberg - Austria
werner.bischof
@fh-joanneum.at

Elmar Krajnc
FH JOANNEUM
Internet Technology
Kapfenberg - Austria
elmar.krajnc
@fh-joanneum.at

ABSTRACT

Today's smartphones are more than just a telephone. With built-in positions sensors a phone can be used as a personal navigation device. Even blind or visual impaired people are used to modern phones. With assisted technologies like VoiceOver or Talkback, blind people are not reliant on phones with haptic/touchable keys. A lot of effort was made to improve the positioning indoor and outdoors, but affordable technologies are not accurate enough to navigate blind persons on a safe trip. The positioning should be improved by telling the user the surrounding environment. In example the user can verify the position of the sidewalk by telling him the location of the street and the buildings. The goal is the selection of a routing maschine which can deliver a safe route and additional trip information (i.e. crossings with acoustic traffic lights).

Keywords

GIS, Routing, Navigation, Blind, Pedestrian

1. INTRODUCTION

For visually impaired and blind people the fastest or the shortest route is not the important fact, they prefer the safest route from a start point to an end point (In figure 1 there are also other user-groups, which prefer specialized routing). There are some navigation systems for blind users available on the market, but no system makes much use of additional data for blind users. OpenStreetMap contains very useful information for visually impaired and blind people. It make it possible to find a safe route from a start point to an end point, which includes acoustical traffic signals and safe sidewalks. Also additional information can be used for navigation. If the user is told that the building is on the right hand side, he/she can verify the information and improve the positioning. If there is a dangerous situation(i.e. the proximity of a waste basket, a poller or a hydrant) a warning should occur.
Today there are some mature open source routing engines available. Many of them focus on car navigation, bicycle navigation and pedestrian navigation. Some of them even include multi-modal routing in combination with public transportation systems. In this paper a comparison between different open source routing engines (PgRouting,OpenTripPlanner and OpenSourceRoutingMachine) is made. Is it possible to transfer the technology from car navigation to blind pedestrian navigation?

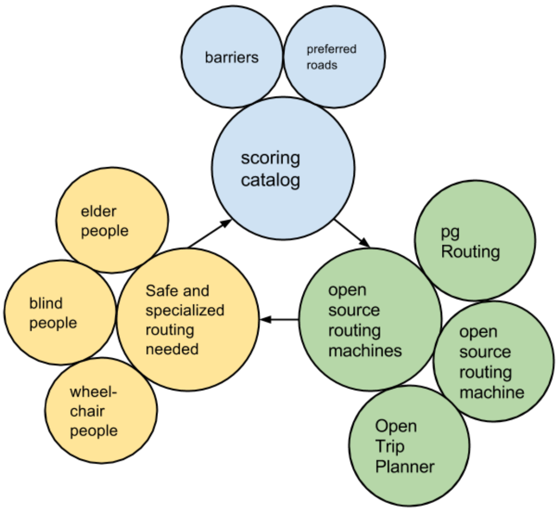


Figure 1: Specialized routing

2. USER REQUIREMENTS

The navigation device should be a smartphone. The user group often has less money and could not afford additional devices. The user interface on the smartphone should be barrier free and useable for blind and visually impaired people. During the navigation process the hands should be free for the blindman stick an additional stuff. The user should feel less disturbance and the surrounding traffic and the enviroment should be recognisable.

3. RELATED WORK IN BLIND PEDESTRIAN ROUTING AND NAVIGATION

The pioneer in free GPS navigation software is Loadstone GPS[2]. It is running on Symbian mobile phones and uses an external bluetooth GPS receiver. The product is very popular in the community of blind and visually impaired people. Nevertheless there is no active development anymore.
On the iOS platform Blindsquare[1] is a very popolar App. It uses point of interest information from foursquare and OpenStreetMap and allow navigation to the PointOfInterests. The actual price is about 24 US Dollar.
OsmAnd (OSM Automated Navigation Directions) is a map and navigation application with OpenStreetMap (OSM) data. All map data can be stored on your device's memory card for offline use. Via your device's GPS, OsmAnd offers routing, with optical and voice guidance, for car, bike, and pedestrian. It has a Turn-by-turn voice guidance and does an automatic re-routing whenever you deviate from the route. OsmAnd works well for blind people with Talkback when the uses pre stored favourites. The greatest challenge for the blind people is the destination input, which is only possible with standard keyboard.
'Ready.Set.Access'[7] is a master thesis about intermodal door-to-door application for people with disabilities. The implementation was done with the OpenTripPlanner (section 7.3) Routing Software. It combines indoor and outdoor navigation. The main focus is on wheelchair accessibility. The project also analyses the topography of a region and flat slops are preferred. In the thesis a geospatial barrier catalogue with six categories (Way Types, Surface Types, Way Related Attribues, Node Information, General Attributes and Public Transport) was made.
Sven Leitinger[4] from Salzburg Research wrote a guideline to enhance the OSM Data with blind related attributes. He explained most of the OSM tag, which are useful for blind people.
The scientific paper[6] RouteCheckR focus on route finding for mobility impaired people. A safety factor was embedded in the route calculation. It is recommended to read this paper, if you are interested in routing algorithms with includes safety parameters. The algorithms itself are not scope of this work.

4. OPENSTREETMAP (OSM)

OpenSteetMap was founded in the year 2004. In this last 10 years a enormous map material was generated by volunteers. The map data is free available without any licence cost. Therefore a lot of open source routing tools work with OSM data.
The map network consist of nodes and ways like a routing graph with special attributes called map features. There are confirmed map features and proposed map features. Both will be used in the routing process to find the safest route. It is not scope of the project to find new OSM features for blind pedestrians.

4.1 Interesting features for blind people

Some of the main attributes for blind pedestrians. The information should be given to the users during the navigation.

- highway=pedestrian,living_street,path, footway,steps,elevator
- surface=paved,unpaved,asphalt,concrete,concrete:plates, fine_gravel,metal,paving_stones
- footway=sidewalk,kerb,tactile_paving
- crossing=taffic_signals:sound

5. ROUTING PROBLEM

The main task of an ordinary routing system is to find the shortest route or the fastest route. In blind person routing also safety plays a important role. In general those five requirements[7] are needed to do navigation and routing.

1. A procedure to set start point and end point
2. A connected network (routing graph), on which the routing can be performed
3. An algorithm that computes the route between start point and end point
4. The inclusion of user requirements in the route computation and selection
5. The presentation of the results to the use

This work focus most on point 4 of the list. The influence of the user requirements on the route.

6. PREREQUISITES

Following questions has to be asked to choose the right routing service for the comparison.
Is the routing service free of charge and is it possible to adopt it for blind pedestrian routing? Is it possible to put additional information in the routing graph? Can OpenStreetMap data be used to generate the routing graph? What hardware is needed to run the routing server? A Web-Frontend is not required for the blind users, but it is important for the developer. It is easier to test different routes and it gives you a good impression about the output of the routing service.

7. COMPARISON BETWEEN DIFFERENT ROUTING TOOLS

In a first selection three tools has been found, which fulfil the requirements. For sure there are more tools around, but these three are well documented and active.

- PgRouting
- OpenTripPlanner
- OpenSourceRoutingMachine

7.1 Generation of the routing graph

All routing services in this comparison use OpenStreetMap data to generate the routing graph. It is important that the routing graph has all the OpenStreetMap attributes, which are important for blind people. The process is illustrated in Figure 2.

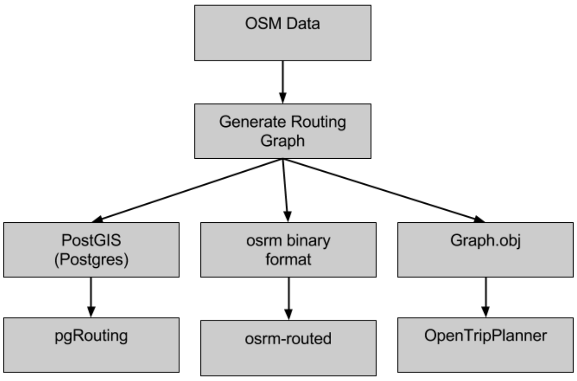


Figure 2: Overview

7.2 pgRouting

pgRouting[3] is an extension to the PostGIS standard. PostGIS brings spatial information to Postgres database. In a well documented workshop from pgRouting the combination of Geoserver and OpenLayers is shown. Also a example with QGIS is included.
It is easy to get a GeoJson track with some server side programming language (i.e. PHP or Perl). So even interfaces to smartphone applications should be manageable. But the lack of turn instructions makes the difference to other routing tools.

+ routing network or route can be displayed with GIS software (i.e. QGIS)

+ many different routing algorithms (Dijkstra, A-Star and Shooting Star) are implemented

+ standardised routing graph (PostGIS)

+ data is easy accessible

- a lot of components (Postgres, PostGIS, osm2pgrouting, GeoServer, OpenLayers) from different vendors

- difficult to set up the environment

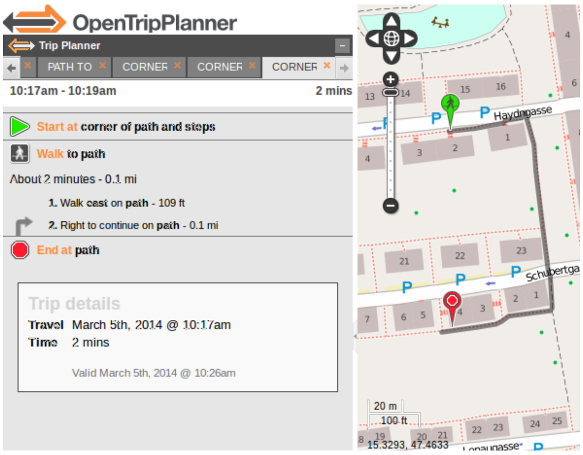
- no turn instructions implemented

- no mature Web-Frontend

The OSM data can be transferred to Postgres database by the program osm2pgrouting. Priorities of roads can be set in a Xml configuration file (Listing 1).

Listing 1: osm2pgrouting Mapconfig.xml

```
<?xml version=”1.0” encoding=”UTF−8”?>
<configuration>
  <type name=”highway” id=”1”>
    <class name=”motorway” id=”101”
        priority=”1.0” maxspeed=”130” />
    <class name=”primary” id=”106”
        priority=”1.15” maxspeed=”90” />
    <class name=”residential” id=”112”
        priority=”2.5” maxspeed=”50” />
    <class name=”living_street” id=”113”
        priority=”3” maxspeed=”20” />
  </type>
</configuration>
```



10:17am - 10:19am 2 mins

Start at corner of path and steps

Walk to path

About 2 minutes - 0.1 mi

1. Walk east on path - 109 ft

2. Right to continue on path - 0.1 mi

End at path

Trip details

Travel March 5th, 2014 @ 10:17am
Time 2 mins

Valid March 5th, 2014 @ 10:26am

Figure 3: Screenshot OTP - Kapfenberg

7.3 OpenTripPlaner (OTP)

The OpenTripPlaner servlet runs on a Java web server. It is a complete bundle which includes a Rest Api and a Web-Frontend(Figure 3). Over the Web-Frontend a start and an end point can be set and with a button click the route can be requested. To Drag-and-Drop the markers as with the OSRM Web-Frontend is not possible. The route request takes much more time as with OSRM. For other Applications like smartphone applications there is a well dokumentated Rest Api, where a route can be requested with many parameters. There is already a open source Android application interacting with the routing server. But with no user interface for blind people.
One routing process can handle multiple profiles (i.e. Walking, Bicycle and Car) and intermodal transportation. GTFS (General Transit Feed Specification) format from Google is used to query the timetables from public transportation companies.

+ includes public transportation (GTFS)

+ intermodal (i.e. combine walk and public transport) trips

+ well documented Rest Api

+ separated Java web-apps (Web-Frontend and routing service)

+ less dependencies (any Java web server)

+ multiple profiles (Bike, walk, transit) handled by a single routing process

+ walk speed selectable

+ safety options in the configure file available

+ easy configuration with Xml file

+ Logging and Debugging

+ open source Android application

- slow route calculation

- need much resources to build routing graph

The graph builder is controlled by an XML (Listing 2) configuration. For medium regions (i.e. the Country Austria) a lot of RAM is needed. OpenTripPlanner is intended for small and medium geographic areas, but with huge amount of RAM also big areas are possible. The safety parameter was invented for bicycles. There can be a notification, if the osm key surface has a specific value (i.e. 'Caution: muddy!'). The safety value of 1 is default. Very safe streets can have a value below 1 and for dangerous streets a value more than 1 can be given. The distance of a street segment will be multiplied with the safe value. A ten times more dangerous street will have the the length of the street times 10.
With a permission key all transportation types (i.e. Pedestrian, Bicycle, ..) can be put on a white list. So street segment can be accessed by this type of transportation.
It should be possible to convert the bicycle safety features to pedestrian safety features.

Listing 2: OTP GraphBuilder.xml

```
<bean class="WayPropertyPicker">
 <property name="specifier" value="highway=
     motorway;cycleway=lane" />
  <property name="properties">
   <bean class="WayProperties">
    <property name="safetyFeatures" value="
        1.5,1.5" />
    <property name="permission" value="
        BICYCLE_AND_CAR" />
   </bean>
  </property>
</bean>
```

7.4 OpenSourceRoutingMachine (OSRM)

The project is separated in the routing service and a Web-Frontend(Figure 4). The routing service is written in C++ and needs less hardware resources then OpenTripPlanner. The routing graph can be influenced by a profile configuration. The profile is a lua script file. So even more sophisticated configurations can be done, but it is more complex as a Xml configuration file. In the standard configuration no safety properties are in the profile file. The route can be influenced by the speed parameter, so a safe street can have a higher speed value. For sure this will also change the travel time. For every profile a single instance of the routing machine is needed.

+ very fast[5], need less resources

+ lua script language to configure the routing graph

+ shortest path calculation contraction hierarchies, faster then dijskta algorithm

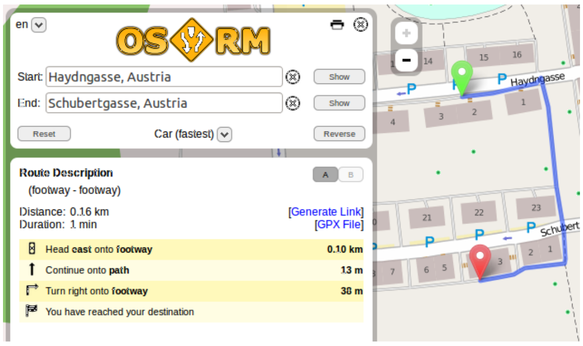
+ blacklist of street, which should not be used

+ delivers Gpx track or turn instruction list in Json

+ nice Web-Frontend

+ supports multiple targets in the route request

- no safety properties available



Start: Haydngasse, Austria
End: Schubertgasse, Austria
Car (fastest)

Route Description
(footway - footway)
Distance: 0.16 km
Duration: 1 min

Head east onto footway 0.10 km
Continue onto path 13 m
Turn right onto footway 38 m
You have reached your destination

Figure 4: Screenshot OSRM - Kapfenberg

Table 1: Example Routing List

No	Turn Angle	Road Name/Type	Distance	Heading
4	282	Werk-VI-Strasse	250m	NW
12	14	Path	24m	SW

The route can be configured for blind pedestrians, but there will be no turn instruction if the road name/type stays the same. So even on an orthogonal crossing there can be no turn instruction, if the road stays the same. The missing turn instruction can be found on Figure 4. There is only one turn instructions, instead of two. The same problem occour in the OpenTripPlaner (Figure 3).
The route response is well documented. It delivers all track points and for some track points a turn instruction. Some example turn instructions can be found on Table 1.
The route request is handled by a Plug-in. New features can also be added by a Plug-in.

In the profile settings (Listing 3) different speeds for all kind of roads can be set. Additional OSM meta data can be delivered in the road name/type field. On every different road name/type a turn instruction is triggered.

Listing 3: OSRM profile.lua

```
walking_speed = 5

speeds = {
   ["residential"] = walking_speed*0.5,
   ["living_street"] = walking_speed,
   ["track"] = walking_speed*0.8,
   ["path"] = walking_speed*0.8,
   ["steps"] = walking_speed*0.5,
   ["pedestrian"] = walking_speed,
   ["footway"] = walking_speed
}

surface_speeds = {
   ["fine_gravel"] =    walking_speed*0.75,
   ["gravel"] =         walking_speed*0.75,
   ["sand"] =           walking_speed*0.5
}
```

8. SYSTEM DESIGN

The system was designed as Server/Client Infrastructure (Figure 5). The network graph is held on the server and the routing calculation is also done on the server. The client ask for the best route from start point to a destination point. Even multiple destination can be handled by the server. Then the server replies with a Json formated file. The Json

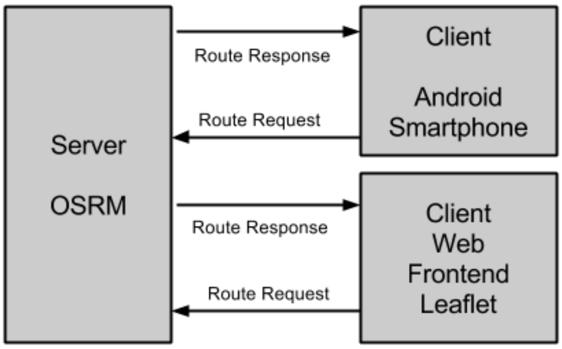


Figure 5: System Design

file contains the Gpx Track and all turn instructions. The turn instructions contain the turn angle, the street name or road type, the distance to the next turn instruction and the heading information (Table 1). A network connection is needed for every route request.

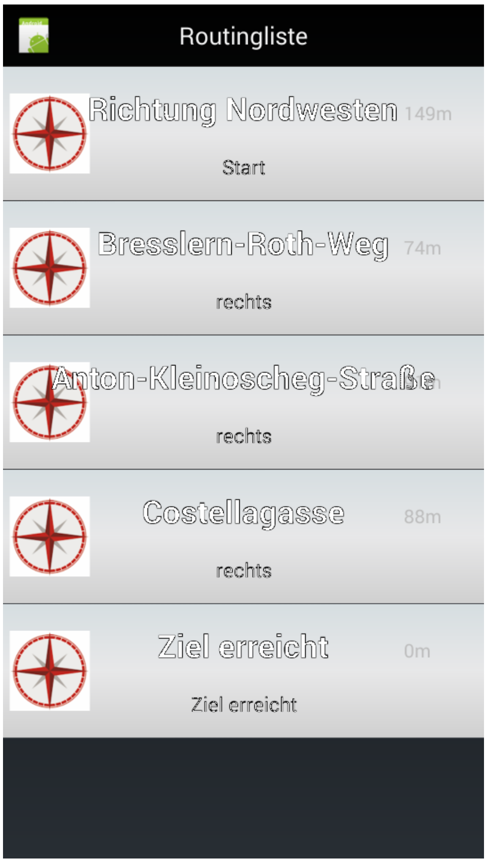
9. FIRST RESULTS AND TESTS

The Android application development started at the same time as the comparison between the different routing machines. It was clear that the Application request a route with a start and a stop position and the server replies with a well structured (i.e. Xml or Json) format.
The user interface and the navigation instructions are much more important on the smartphone. The routing server should be replaceable by any routing service in the comparison. By the way OSRM was chosen as routing service for the first test. On the Application side a target position can be selected from a list and the actual position was used as start position. After the routing request, the response is displayed in a new list (Figure 6).
The user can explore the whole trip on the screen. Turn instruction by turn instruction. The distance to the next crossing point is spoken periodically. If the proximity of the crossing point was reached the turn instruction was told.

10. CONCLUSION

All three software products can be adopted for blind pedestrian routing, but the implementations differ a lot. So a question can be which programming language (Java, C++ or SQL) is your favourite. The main focus of every implementation is unique and the systems differ a lot.
OpenTripPlaner is the only Software with a safety parameter and has the most features. On the other hand the package needs a huge amount of resources.
OpenSourceRoutingMaschine performs very good and supports flexible configuration options. But to make use of all the function a good knowledge of the software structure is needed.
PgRouting use a standardized road network (Postgis) and has more focus on different routing algorithms. A lot of software development will be needed to implement the missing functions.

11. FUTURE WORK


Routingliste

Richtung Nordwesten 149m
Start

Bresslern-Roth-Weg 74m
rechts

Anton-Kleinoscheg-Straße
rechts

Costellagasse 88m
rechts

Ziel erreicht 0m
Ziel erreicht

Figure 6: Android App

The first version was just a prove of concept. The main focus was on the user interface on the smartphone side and on a stable server and client communication. The navigation has to be improved a lot. There has to be a mode where blind people can follow a track in a park without much orientation kerbs. Also a module is needed to get information about the near environment (i.e. Position of the road and the buildings). This information could not be found in the route and has to queried frome the data source itself. There have to be turn instructions where the road name/type stays the same, but makes a curve (i.e. priority roads). And an evaluation of the priority of foot ways has to be done.

12. REFERENCES

[1] Blindsquare webpage. www.blindsquare.com/about. Accessed: 2014-02-05.

[2] Loadstone webpage. www.loadstone-gps.com. Accessed: 2014-02-05.

[3] Pgrouting webpage. www.pgrouting.org. Accessed: 2014-02-05.

[4] S. Leitinger. *Leitfaden zur OSM-Datenerfassung fuer blinden-spezifisches Routing*, 2013.

[5] D. Luxen and C. Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 513–516, New York, NY, USA, 2011. ACM.

[6] T. Völkel and G. Weber. Routecheckr: Personalized multicriteria routing for mobility impaired pedestrians. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 185–192, New York, NY, USA, 2008. ACM.

[7] T. Weyrer. Ready.set.access - development of an intermodal door2door routing application for people with physical impairments. Master's thesis, 2013.