# Open-Source-Erreichbarkeitsanalysen für den Öffentlichen Verkehr

Ersatzvortrag FOSSGIS 2018

Charlotte Pusch, M.Sc. Ole Röntgen, M.Sc.

Technische Universität Hamburg
Institut für Verkehrsplanung und Logistik



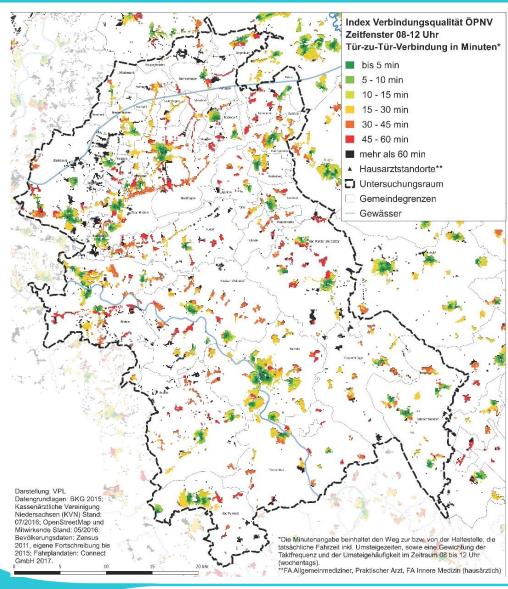
### **UrbanRural SOLUTIONS**



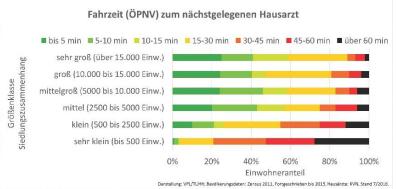
renddarstelli Regional Organisatorische Einbindung Intuitiv KleinräumigesBevölkerungsmodellE StatistikenBevölkerungsfortschreibung Status Quo Visualisierung Bedürfnisorientiert Verkehrsmodell Pflege Mutzen nnovationsprozess Automatisiert VerstetigungInforn richtungenTransparent Datenban Kapazitäten

# Da wollen wir hin...





ÖPNV-Erreichbarkeit des nächstgelegenen Hausarztes, Stand 2016 Schaumburg/ Hameln-Pyrmont







### Was steckt in dieser Karte...





## Kleinräumiges Bevölkerungsmodell

- Einwohner auf 100x100 Metern
- Status Quo
- Trend



### Räumliches Standortmodell

- Adressgenaue
   Geodaten
- Status Quo
- Planfall

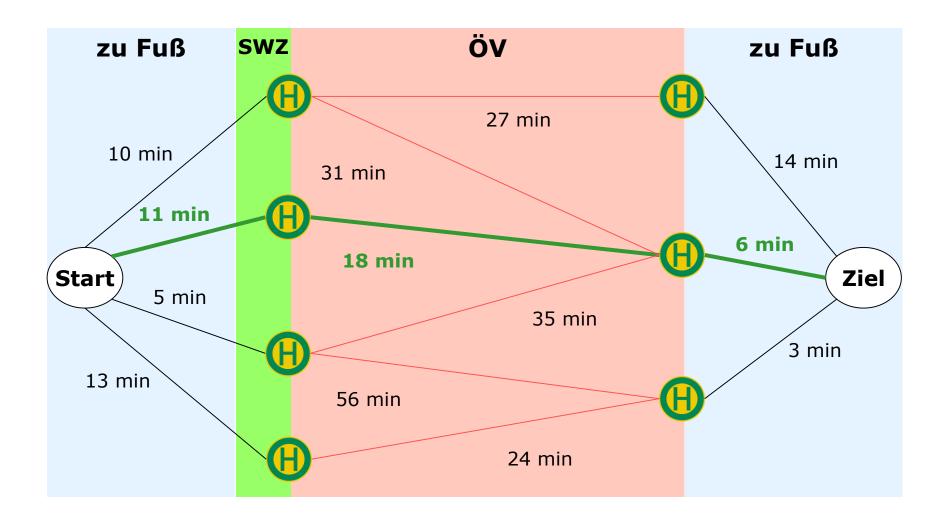


#### Verkehrsmodell

- IV & ÖV
- Reisezeit und Realdistanz
- Status Quo



# Herausforderung Routenberechnung ÖV





## Was brauchen wir?



# Ein Tool, das

- eine große Anzahl an Routen schnell berechnet und dabei
- OSM-basierte IV-Netze (Fuß-/Rad-/Autoverkehr) und
- ÖV-Fahrplandaten in gebräuchlichen Datenformaten verarbeiten kann.

# **OpenTripPlanner**



- Open Source, freie Lizenz (GPL)
- Beherrscht multi- und intermodales Routing
- Datenbasis: OSM für Individualverkehr sowie GTFS für ÖV
- Programmiert in Java
- Per Jython (u.a.) scriptbar
- Möglichkeit zu many-to-many-Analysen



# **Exkurs: GTFS**



GTFS ist ein offenes Datenformat für ÖV-

Fahrplandaten

Leicht les- und verarbeitbar

Immer weitere Verbreitung



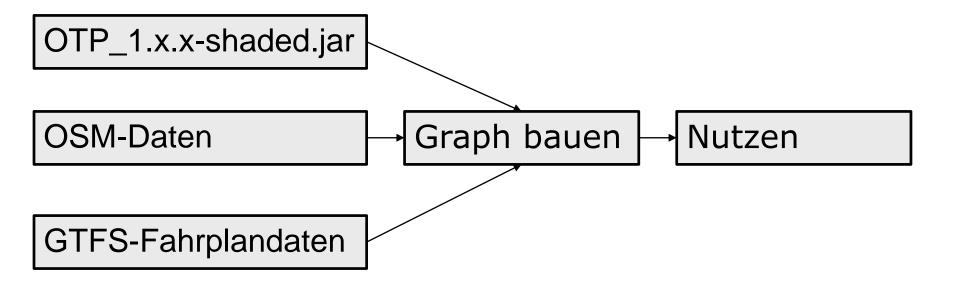
November 2017, Quelle: rettedeinennahverkehr.de



# **OTP: Installation**



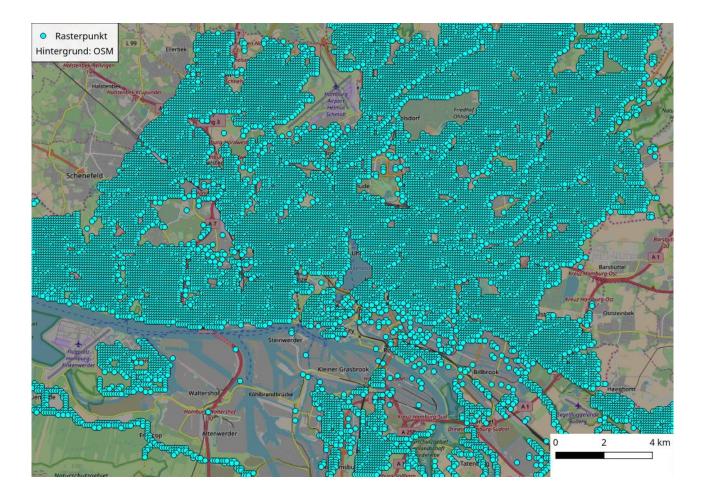
Vorbereitende Schritte OTP



# **OTP: Gefüttert mit**

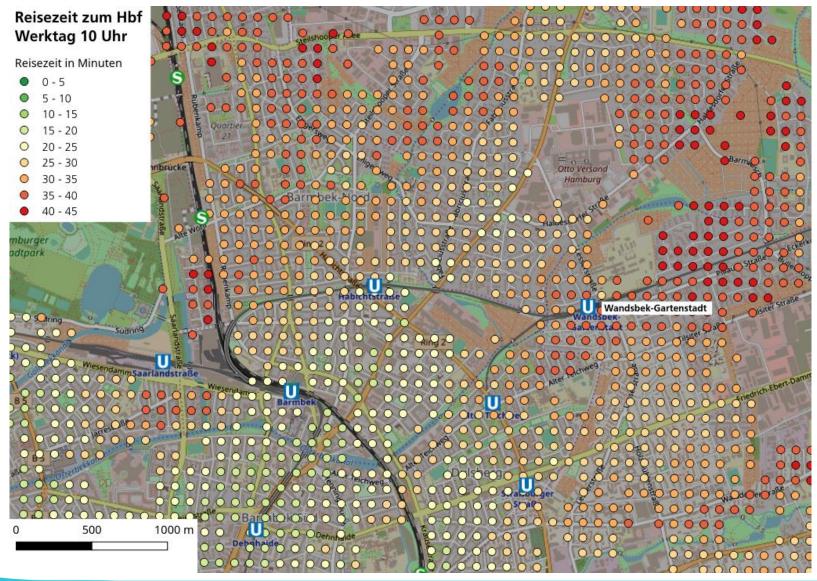


Beispiel Input: Rasterpunkte (100 m)



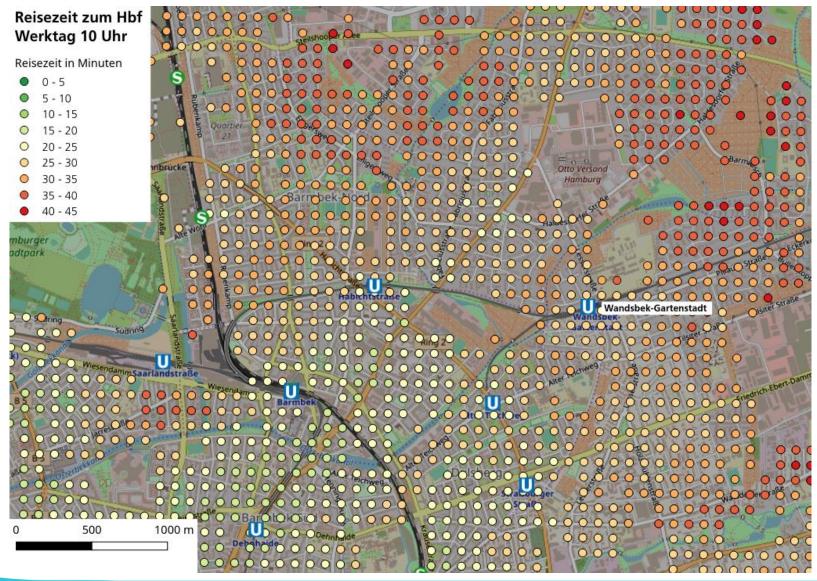
# **Unerwartete Ergebnisse**





# **Unerwartete Ergebnisse**





# Lösungen...

```
// Model parameters are here. //
// Constants for when there is a traffic light.
/** Expected time it takes to make a right at a light. */
private Double expectedRightAtLightTimeSec = 15.0;
// OR: Anpassung, war: 15.0
/** Expected time it takes to continue straight at a light. */
private Double expectedStraightAtLightTimeSec = 15.0;
// OR: Anpassung, war: 15.0
/** Expected time it takes to turn left at a light. */
private Double expectedLeftAtLightTimeSec = 15.0;
// OR: Anpassung, war: 15.0
// Constants for when there is no traffic light
/** Expected time it takes to make a right without a stop light. */
private Double expectedRightNoLightTimeSec = 8.0;
// OR: Anpassung, war: 8.0
/** Expected time it takes to continue straight without a stop light. */
private Double expectedStraightNoLightTimeSec = 5.0;
// OR: Anpassung, war: 5.0
/** Expected time it takes to turn left without a stop light. */
private Double expectedLeftNoLightTimeSec = 8.0;
// OR: Anpassung, war: 8.0
@Override
public double computeTraversalCost(IntersectionVertex v, StreetEdge from, StreetEdge to, TraverseMode mode,
                                   RoutingRequest options, float fromSpeed, float toSpeed) {
    // If the vertex is free-flowing then (by definition) there is no cost to traverse it.
    if (v.inferredFreeFlowing()) {
        return 0;
    // Non-driving cases are much simpler. Handled generically in the base class.
```

# **OTP: Fazit**



- Großes Potential für schnelle Berechnung von OV-Routen auf GTFS/OSM-Basis
- Im Prinzip das, was man für Erreichbarkeitsanalysen braucht.
- Allerdings in der Praxis noch Fallstricke, nicht ohne weitere, komplexe Anpassungen einsetzbar.
- Modifikationen werden anhand von Praxisanforderungen potentieller Nutzer vorgenommen



# **Ausblick**



- Nutzungsmöglichkeit: Szenarien (historische und zukünftige (planned) Daten)
- Andere Möglichkeiten/Tools zum ÖV-Routing?
- Cloud-Dienste





## **Vielen Dank**







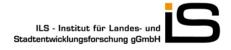












AKADEMIE FÜR RAUMFORSCHUNG UND LANDESPLANUNG

