

Erstellung eines Routing-Profiles auf Basis von OSM / Öffentlichen Daten für Feuerwehrfahrzeuge

Amandus Stefan Butzer

23. September 2017

Kurzfassung

Deutsche Kurzfassung

Abstract

Abstract in Englisch

Erklärung

Hiermit erkläre ich

Danksagungen

An dieser Stelle möchte ich all denen Danken..

Inhaltsverzeichnis

1	Einleitung	2
1.1	Motivation	2
2	Theoretische Grundlagen	2
2.1	Graphen	2
2.2	Routing	3
2.3	Isochronen Berechnung	4
3	Generierung des Routing-Profiles	5
3.1	Informations Erhebung	5
3.2	Aufbau graph backend	5
3.3	Limitierende Faktoren	5
3.4	Erweiternde Faktoren	5
4	Ergebnisse	5

5	Fazit	5
6	Future Work or Ausblick	5

1 Einleitung

Zum Zeitpunkt der Erstellung dieser Arbeit ist der Verfasser in der Geoinformatik Abteilung des Geographischen Instituts der Ruprecht-Karls-Universität Heidelberg als wissenschaftliche Hilfskraft des openrouteservice (ORS) tätig. Der ORS bietet neben Geocoding, Routing und Location Service auch einen Isochrone Service an. Immer wieder wurden Anfragen bezüglich Erreichbarkeitsanalysen aus dem Rettungs- und Brandschutzwesen erhalten. Für Polizei, Rettungsdienst und Feuerwehr geht es vor allem um das Einhalten amtlich vorgegebener Hilfsfristen. Diese stellen die wichtigsten Eigenschaften für die Planung und Qualität der Einsätze von Feuerwehr und Rettungsdienst dar. Der Brandschutz ist im Gegensatz zum Rettungsdienst eine kommunale Aufgabe und unterliegt nur in manchen Bundesländern bestimmten Standards Stein 2015.

Da mit dem Isochrone-Service des ORS Erreichbarkeitsanalysen durchgeführt werden, ist dieser für die Erstellung eines Brandschutzbedarfsplans der Feuerwehr geeignet. Jedoch kann der Dienst in seiner bisherigen Form noch nicht alle erforderlichen Anforderungen für Einsatzfahrzeuge erfüllen.

In dieser Arbeit wird daher ein Emergency-Routing-Profil in Kooperation mit der Freiwilligen Feuerwehr Lützelburg entwickelt. Die Implementierung ist auf eine Fahrzeugklasse der Feuerwehr begrenzt. Allerdings wird bei der Erstellung des Emergency Profils darauf geachtet, dass Erweiterungen für diverse Einsatzfahrzeuge sehr einfach möglich sind. (Aufgrund des geringen Umfangs einer Bachelor Arbeit kann das in dieser Arbeit jedoch nicht erarbeitet werden.)

Grundsätzlich basiert das Profil auf dem Backend des bereits bestehenden Routing Service des ORS. Zusätzlich werden Java Funktionen implementiert, die speziell auf das Emergency Profil zugeschnitten sind. Zur Darstellung wurde das ORS Frontend mit Java-Script angepasst. Dadurch können die Ergebnisse in verständlicher und anschaulicher Weise dargestellt werden. (weitere Infos über technic details?)

1.1 Motivation

STVO Anwendbar für fast alle Einsatzfahrzeuge

2 Theoretische Grundlagen

Als Grundlage für die Berechnung kürzester Wege bedienen wir uns eines Speicherform der Mathematik, dem Graphen.

2.1 Graphen

Ein Graph ist ein sehr nützliches Konzept um Objekte und deren Verbindungen untereinander zu modellieren. eine Sammlung von Punkten, bei denen einige Knotenpunkte durch Kanten verbunden sind. Kurt Mehlhorn 2008, S. 49 (Graph definition with cite)

(simple graph picture)

Der Vorteil von Graphen ist die einfache Darstellung. Die Knoten eines Graphen können einfache, mehrfache oder gar keine Kanten haben. Darüber hinaus können sie mit sich selbst verbunden sein und eine Schleife bilden. Außerdem können zwei Punkte mit mehreren Kanten verbunden sein. Das erfüllt bei einem ungerichteten und ungewichteten Graphen allerdings keinen Zweck, da eine Kante für das Anzeigen der Relation ausreichend ist(richtig?).

Es sollte aber nicht vergessen werden, dass ein Graph nicht die räumliche Position der Objekte sondern nur ihre Relation zueinander ausdrückt! Es können also zwei Graphen unterschiedlich angeordnet sein und dennoch einander entsprechen. Man spricht hier von isomorphen Graphen **zitat**.

Gerichtete Graphen Für das Routing ist es wichtig von einem Knoten zu einem anderen zu gelangen. Im Gegensatz zu einem ungerichteten Graphen können bei einem gerichteten Graphen Kanten nur in einer Richtung durchlaufen werden. Die Kanten werden daher durch Pfeile anstatt Linien dargestellt.

(Abb gerichtet Graph)

Gewichtete Graphen In dieser Arbeit bezeichnet der Begriff "gewichteter Graph" einen Kanten-gewichteten Graphen, bei dem jeder Kante ein Wert zugewiesen wird. Neben dem Kanten-gewichteten gibt es auch Knoten-gewichtete Graphen, bei welchen entsprechend die Knoten gewichtet werden. Diese werden aber nur für wenige Problemstellungen gebraucht und sind hier nicht von Belang. Gewichtete Graphen können gerichtet und ungerichtet sein. Ein klassisches Beispiel hierfür ist der Linien-Netzplan einer Bahn, bei dem die Knotenpunkte einzelne Haltestellen darstellen und die Kantengewichte die benötigten Minuten beinhalten.

(Abb gewichtet Graph)

Mit gewichteten Graphen können diverse Problemstellungen gelöst werden, zum Beispiel die Bestimmung maximaler (Durch-)Flüsse in Rohrsystemen oder das Berechnen kürzester Wege.

2.2 Routing

Routing ist ein Begriff aus der Nachrichtentechnik und bezeichnet die Bestimmung des günstigsten Weges für die Übertragung von Daten in einem Netzwerk (cite DUDEN). In der Navigation bedeutet Routing ebenfalls das Ermitteln eines besonders günstigen (häufig des schnellsten) Weges von Startpunkt A zu Zielpunkt B. Das Netzwerk in dem dieser Weg ermittelt werden soll ist in diesem Fall ein gerichteter und gewichteter Graph, der ein reales Straßen- oder Wegenetz repräsentiert.

Wenn die Kante eines solchen Graphen also eine Straße darstellt, beinhaltet das Kantengewicht dabei die Distanz. Die Richtung der Kante ist äquivalent zur erlaubten Fahrtrichtung zum Beispiel im Fall von Einbahnstraßen oder Autobahnen.

Eine der naheliegendsten Problemstellungen ist das *shortest path problem* für ein Paar von Knotenpunkten. Dabei wird der kürzeste Weg von einer Ecke zu einer anderen Ecke des Graphen gesucht. bei welchem der kürzeste weg von einer Ecke zu entweder einer anderen Ecke, allen anderen Ecken

- what is shortest pathing Das Problem des kürzesten Weges
- what algorithms can be used
- algorithmus Dijkstra

+ contraction hierarchies

2.3 Isochronen Berechnung

Isochronen (Herkunft wort iso, chronos)

Wenn in einem gewichteten Graphen die Kanten die benötigte Zeit enthalten um von einem Knoten zum nächsten zu gelangen, können damit Isochronen berechnet werden. Isochronen sind Linien gleicher Zeit. Für die Berechnung wird ein Zentrum und das Zeitlimit benötigt. Das resultierende Objekt ist ein Polygon, welches jeden in gegebenem Zeitlimit erreichbaren Punkt beinhaltet. Isochronen können auf unterschiedliche Arten berechnet werden.

Benutzt routing

+ shaping (darstellung)

marching squares grid based (recursive grid?), delauny erklären shape based(triangles) + unsere implementierung shape based (points)

Gitterbasierter Ansatz: Beim *marching squares* algorithmus wird um das Zentrum ein Gitter über dem Graphen gebildet. Die Eckpunkte des Gitters erhalten dabei die Werte des nächsten Punktes auf dem Graphen. Anschließend werden auf den Kanten des Gitters diejenigen Punkte markiert, bei denen der Wert mit dem gesuchten Zeitlimit übereinstimmt. Die markierten Punkte werden verbunden und bilden schließlich die Isochrone.

(Abb. recursive grid / marching squares)

Der Vorteil dieses Algorithmus ist, dass die Maschengröße des Gitters angepasst werden kann. Bei sehr kleinen Maschen liefert der Algorithmus ein sehr genaues Ergebnis. Allerdings werden dabei mehr Ressourcen zur Berechnung gebraucht. Daher können nur kleine Gebiete und geringe Zeitlimits berechnet werden. Bei weiten Maschen ist der Algorithmus dagegen sehr schnell und kann große Distanzen und lange Zeitspannen berechnen. Das Ergebnis ist dementsprechend aber auch ungenauer.

TIN: Delaunay Triangulation TODO

Formenbasierter Ansatz: Die Implementierung des ORS zur Berechnung von Isochronen verwendet ist ein formen basierter Ansatz (wiederholung :/). Zuerst werden mit dem in Kapitel .. bereits ausführlich erklärten Dijkstra Algorithmus alle in gegebener Zeit erreichbaren Kanten markiert. Anschließend werden die geographischen Punkte der Geometrie der Kante extrahiert. Jede Kante des Graphen stellt ein Wegstück der realen Welt dar. Im Graphen können Kanten allerdings nur durch gerade Linien dargestellt werden. Die wirkliche Geometrie ist deswegen auf der Kante gespeichert. Um jeden der extrahierten Punkte wird ein kreisförmiger Pufferbereich gelegt. Dadurch können nahe beieinanderliegende Punkte übersprungen werden. Mit den verbleibenden Punkten wird eine Punktwolke generiert. Auf diese Punktwolke wird nun der alpha shape algorithmus(Zitat) angewandt um die Isochrone als Hülle um die Erreichbaren Wegsegmente zu zeichnen. Durch den Dijkstra Algorithmus können mit diesem Ansatz Distanzen bis zu 100km berechnet werden. Je nach Transportmittel entstehen also Unterschiedliche Zeitlimits für die Isochronen Berechnung: Mit dem Auto können bis zu einer Stunde, mit dem Rad bis zu fünf Stunden und zu Fuß sogar bis zu 20 Stunden abgefragt werden. Dieser Ansatz liefert präzise Ergebnisse und

Shape based: - Uses Dijkstra algorithm to mark all reachable Edges - Extract points from edge geometry(real world geom, becaus in graph we only have simple

straight lines between nodes, geometry is saved in the graph edge as parameter)
- invisible bubbles (buffer) will be drawn around points so close points can be skipped - build a point cloud with remaining points - use Alpha shapes algorithm to draw isochrones around the point line

Distance limitation 100km. resulting in different time limitation for different profiles (1hour car , 5hours bike, 20 hours foot)

Advantage: accurate, fast computation time Disadvantage: Alpha shape library -> no holes.

3 Generierung des Routing-Profiles

3.1 Informations Erhebung

Fragebogen für Feuerwehr Lützelburg¹

3.2 Aufbau graph backend

weightings speed maps

first tests, feedback from firebrigade way too far.

calc weight for dijkstra calc millis for time add up

3.3 Limitierende Faktoren

3.4 Erweiternde Faktoren

4 Ergebnisse

Vergleiche zwischen Firetruck - Emergency Vehicle - Car - Heavy Vehicle + exemplarische reale Beispiele!

Hier ein paar räumliche Beispiele aussuchen und exemplarisch zeigen (Routing und Isochronen), welche Änderungen das Profil mit sich bringt, einerseits innerstädtisch, andererseits auch außerhalb der Stadt. Denn Änderungen als solches ist bisschen schwierig zu definieren. die Jungs aus Lützelburg fragen, welches Gebiet mit den bereits vorhandenen Profilen wirklich schlechte Ergebnisse bringt und jetzt mit Emergency weitaus realitätschere!

5 Fazit

Much more accurate than previous profile.

6 Future Work or Ausblick

Suche nach Löschwasser quellen um den Zielpunkt (osm tag emergency=fire_hydrant) Beschleunigung rush hour , Tag und Nacht Unterscheidung (nachts weniger los auf Straßen Fußgängerzonen ...)

¹Lützelburg ist eine Stadt in Bayern

Literatur

- [Kur08] Peter Sanders Kurt Mehlhorn. *Algorithms and Data Structures*. 2008.
- [Ste15] Jochen Stein. *Qualitätskriterien für die Bedarfsplanung von Feuerwehren in Städten*. 2015. URL: <http://www.agbf.de/pdf/Fortschreibung%20der%20Empfehlung%20der%20Qualitaetskriterien%20fuer%20die%20Bedarfsplanung%20in%20Staedten%20Layout%20neu%202016.pdf>.