TU WIEN
DEPARTMENT OF
GEODESY AND
GEOINFORMATION

Pedestrian routing of dynamic areas using Volunteered Geographical Information (OpenStreetMap)

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Kartographie

eingereicht von

Héctor Ochoa Ortiz
Matrikelnummer 12042548

am Department für Geodäsie und Geoinformation

der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.nat. Dr.rer.nat. Georg Gartner
Betreuung: Dr. Anita Graser
Rezensent: M.Sc. Sagnik Mukherjee

Wien, 9. September 2022

Verfasser	Betreuung

TU WIEN
DEPARTMENT OF
GEODESY AND
GEOINFORMATION

Pedestrian routing of dynamic areas using Volunteered Geographical Information (OpenStreetMap)

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Cartography

by

Héctor Ochoa Ortiz
Registration Number 12042548

to the Department of Geodesy and Geoinformation

at the TU Wien

Supervisor: Univ.Prof. Mag.rer.nat. Dr.rer.nat. Georg Gartner
Supervisor: Dr. Anita Graser
Reviewer: M.Sc. Sagnik Mukherjee

Vienna, 9th September, 2022

Author	Supervisor

Cartography M.Sc.

Master thesis

Pedestrian routing of dynamic areas using Volunteered Geographical Information (OpenStreetMap)

Héctor Ochoa Ortiz

Technical University of Munich
TECHNISCHE UNIVERSITÄT WIEN Vienna University of Technology
TECHNISCHE UNIVERSITÄT DRESDEN
ITC UNIVERSITY OF TWENTE.

2022

Statement of Authorship

Héctor Ochoa Ortiz

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:
"Pedestrian routing of dynamic areas using Volunteered Geographical Information (OpenStreetMap)"
I have fully referenced the ideas and work of others, whether published or unpublished.
Literal or analogous citations are clearly marked as such.

Vienna, 9th September, 2022

Héctor Ochoa Ortiz

Danksagung

Danke an meine Betreuer Georg Gartner und Anita Graser, die mir und meinem Projekt von Anfang an vertraut haben und es über 6 Monate betreut und mit Feedback begleitet haben.

Besonderer Dank geht an HeiGIT gGmbH und die Professur für Geoinformatik / GIScience (GIS) an der Universität Heidelberg und insbesondere an Andrzej K. Oleś für die Hilfe bei der Überprüfung und Beantwortung von Fragen zu meinem Code.

Danke an meine Freunde und Familie, die mich auf diesem Weg begleitet und mich als Person verbessert haben.

Acknowledgements

Thanks to my supervisors Georg Gartner and Anita Graser for trusting me and my project from the early stages, and providing supervision and feedback on it over the course of 6 months.

Special thanks to HeiGIT and the GIScience Research Group at Heidelberg University, and specifically to Andrzej K. Oleś, for the help in reviewing and answering questions about my code.

Thanks to my friends and family for accompanying me along the way, and improving me as a person.

Agradecimientos

Gracias a mis tutores Georg Gartner y Anita Graser por confiar en mí y en mi proyecto desde las primeras etapas, y por brindarme supervisión y retroalimentación durante 6 meses.

Un agradecimiento especial a HeiGIT y a la Cátedra de Geoinformática y Ciencia SIG de la Universidad de Heidelberg, y específicamente a Andrzej K. Oleś, por ayudarme a revisar y responder preguntas sobre mi código.

Gracias a mis amigos y familiares por acompañarme en el camino y mejorarme como persona.

Kurzfassung

Im Gegensatz zu Fahrzeugen, treffen Fußgänger Routenentscheidungen nicht nur auf Basis von linearen Objekten. Routing-Netzwerke haben in der Vergangenheit ignoriert, dass Fußgänger einen höheren Grad an Bewegungsfreiheit besitzen und sich frei durch offene Räume bewegen können. Hinzu kommt, dass unsere reale Welt nicht statisch ist, sondern sich dynamisch verändert. Außerdem versagen Routing-Apps oftmals dabei, die unterschiedliche Beschaffenheit der Realität vor Ort abzubilden.

Indem ich sowohl zeitabhängige als auch offene Bereiche kombiniere, kann ich ein komplexes Szenario namens Time Dependent Open Areas (TDOAs) definieren.

Das „Ökosystem“ der OSM-Anwendungen bietet Ausgangspunkt für meine Lösung und ebnet den Weg für die Entwicklung zukünftiger Arbeiten auf einer bereits etablierten Basis.

Ich schlage eine standardisierte Lösung (Schema) zur Kartierung von TDOAs vor, welche sich aus aktuellen OSM-Schemata für Fußgänger- und zeitabhängige Zonen ableitet.

Danach kombiniere ich zeitabhängige und Fußgängerzonen-Routing-Algorithmen, um einen einzigen einheitlichen Routing-Dienst für TDOAs zu erstellen. Der vereinheitlichte Routing-Dienst verwendet OSM-Daten und das neue Schema, um den Routing-Graphen zu generieren.

Diese Arbeit zeigt, dass TDOAs in OSM darstellbar sind. Mein Vorschlag kann sich an viele Realitäten anpassen, bevor diese nicht abgebildet oder berücksichtigt werden würden. Gleichzeitig wird die OSM-Philosophie bewahrt.

Routing ist ebenfalls nachweislich möglich, wenngleich der aktuelle Ansatz nicht perfekt ist. Die Arbeit sollte und muss in zukünftigen Iterationen neu definiert werden, um das Problem angrenzender Bereiche zu lösen.

Mehrere Anwendungsbereiche und zukünftige Arbeiten werden vorgeschlagen. Diese Masterarbeit sollte nur als Sprungbrett in eine breitgefächerte Thematik betrachtet werden, welche noch zu entwickeln ist.

Abstract

Pedestrians, contrasting with vehicles, do not take routes only over defined linear features. Routing networks have historically ignored that pedestrians have a higher degree of freedom, and can naturally move through open spaces. On top of that, our real world is not static, it changes dynamically. Routing apps also fail in most cases to portray the varying character of the reality on the ground.

Combining both time dependent, and open areas I can define a complex scenario called Time Dependent Open Areas (TDOAs).

The “ecosystem” of OSM applications provides a starting point for my solution, and paves the way for the development of derived work on an already established base.

I propose a standard way (schema) of mapping TDOAs, deriving from the current OSM time dependent and pedestrian open areas schemas.

After that, I combine time dependent and open area routing algorithms to create a single unified routing service for TDOAs. The unified routing service uses OSM data and the new schema to generate the routing graph.

TDOAs are demonstrated to be representable in OSM. My suggested proposal is able to adapt to many different realities before not mapped or considered, while still preserving OSM philosophy.

Routing is also proved to be possible, albeit the current approach is not perfect. Work should and must be redefined in future iterations, to fix the adjacent areas border problem.

Several use cases and future work is proposed. This thesis should just be considered as the stepping stone in a much broader topic, yet to be developed.

Resumen

Los peatones, a diferencia de los vehículos, no toman rutas solo sobre características lineales definidas. Históricamente, las redes de ruteo han ignorado que los peatones tienen un mayor grado de libertad y pueden moverse naturalmente a través de espacios abiertos. Además de eso, nuestro mundo real no es estático, cambia dinámicamente. Las aplicaciones de enrutamiento también fallan en la mayoría de los casos al no representar el carácter variable de la realidad sobre el terreno.

Combinando áreas abiertas y dependientes del tiempo, puedo definir un escenario complejo llamado Time Dependent Open Areas (TDOAs).

El «ecosistema» de aplicaciones de OSM proporciona un punto de partida para mi solución y allana el camino para el desarrollo de trabajos derivados sobre una base ya establecida.

Propongo una forma estándar (esquema) de mapear TDOAs, derivada de los esquemas actuales de áreas abiertas para peatones y dependientes del tiempo de OSM.

Después de eso, combino algoritmos de enrutamiento de área abierta y dependientes del tiempo para crear un único servicio de enrutamiento unificado para TDOAs. El servicio de enrutamiento unificado utiliza datos de OSM y el nuevo esquema para generar el gráfico de enrutamiento.

Se ha demostrado que las TDOAs se pueden representar en OSM. Mi propuesta sugerida es capaz de adaptarse a muchas realidades antes no mapeadas o consideradas, mientras conserva la filosofía de OSM.

También se ha demostrado que el enrutamiento es posible, aunque el enfoque actual no es perfecto. El trabajo debe y debe redefinirse en iteraciones futuras, para solucionar el problema de borde de áreas adyacentes.

Se proponen varios casos de uso y trabajo futuro. Esta tesis debe ser considerada simplemente como la primera piedra en un tema mucho más amplio, aún por desarrollar.

Contents

CHAPTER 1

Introduction

1.1 Motivation and problem statement

Contrasting with vehicles, pedestrians do not take routes only over defined linear features. Routing networks have historically ignored that pedestrians have a higher degree of freedom, and can naturally move through open spaces (Graser, 2016). Open Area (OA) routing allows for handling of those suspects.

On top of that, our real world is not static, it changes dynamically. Routing apps also fail in most cases to portray the varying character of the reality on the ground. Time Dependent (TD) routing apps can respond to this.

Combining both time dependent, and open areas I can define a complex scenario called Time Dependent Open Areas (TDOAs). There are no routing services I am aware of that can handle TDOAs.

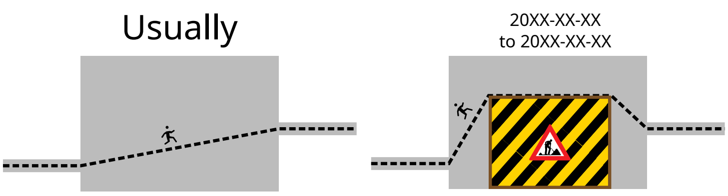


Figure 1.1: Example of a TDOA. A pedestrian square is usually fully traversable. During some days, there is some construction work, so a pedestrian would have to walk around the construction site to reach the opposite end of the square.

When commercial solutions do not exist, or if they are not accessible for organizational, financial or other reasons, Volunteered Geographical Information (VGI), that is edited by individual users, can fill the gap in the availability of digital geographical information (Goodchild, 2007). OpenStreetMap (OSM) is the most renown actor of VGI, being a

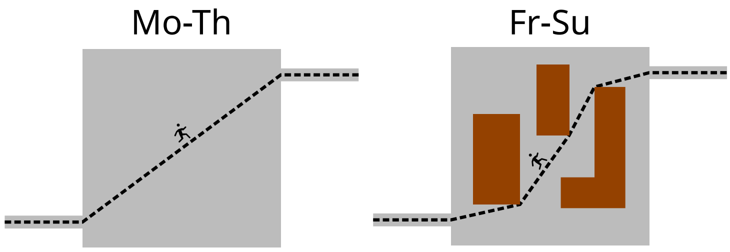


Figure 1.2: Example of a TDOA. A pedestrian square is fully traversable from Monday to Thursday. From Friday to Sunday, there is a weekly market that changes the areas that can be walked over, so a pedestrian would have to walk around the market stalls to reach the opposite end of the square.

free editable spatial database of the world, from which data can also be downloaded and reused with an open license (OpenStreetMap Wiki, 2022a). This openness, in both data and specification, helped with the development of a whole range of applications around it, creating a sort of "ecosystem".

The ecosystem of OSM applications provides a starting point for my solution, and paves the way for the development of derived work on an already established base.

1.2 Research identification

1.2.1 Research Objectives

In this thesis I aim to develop a standard way (schema) of mapping TDOAs, deriving from the current OSM time dependent (OpenStreetMap Wiki, 2022c) and pedestrian open areas (OpenStreetMap Wiki, 2022b, 2022f) schemas. After that, I will combine time dependent and open area routing algorithms to create a single unified routing service for TDOAs. The unified routing service will use OSM data and the new schema to generate the routing graph.

I also aim to develop a web client, that acts as a graphical interface to interact with the routing service. This web client will have the options to set a start and end points, and a time. It will also incorporate a map to visualize the route.

I will prove the tagging schema and the unified routing service in a real world example. The surveyed data will be then incorporated and merged into the existing OSM database, and therefore openly shared for other uses. As OSM already has data contributed over the years, the survey will focus on improving the current data, and adding time dependency to the existing areas.

After that, I aim to analyze the adequateness and effectiveness of the results obtained. A comparison will be made of the routes generated without TDOAs, with only Time Dependent (TD) routing, with only Open Areas (OAs), and with both combined (TD + OAs).

1.2.2 Research Questions

- Q1 How can a new schema for TDOAs be derived from current OSM time dependent and open area schemas?
- Q2 How can we create a routing service that can route through TDOAs?
  - Q2A How can we combine current routing services that can handle time dependent routing over OSM data with those that can handle open areas?
  - Q2B How can we make them work with our new proposed schema?
- Q3 How effective are the proposed schema and routing service in a real world example?
  - Q3A How well can the schema portray the reality in this example?
  - Q3B How does the route quality and accurateness improve over not mapping TDOAs?
  - Q3C How many extra computing resources are needed when mapping TDOAs in comparison to when not mapping TDOAs?

1.3 Thesis structure

This thesis is divided into seven chapters, and is structured as follows.

Chapter 1 is the current chapter, and introduces the research topic to the reader, explaining the motivation behind it and identifying the research objectives and questions.

Chapter 2 dives onto the background of the research topic, reviewing the literature and previous work about it.

Chapter 3 explains the followed methodology and workflow to reach the desired results and answer the research questions Q1 and Q2.

Chapter 4 showcases the real world examples used for the case study, with descriptions and explanations of why they are chosen.

Chapter 5 shows the results of the case study, explaining the tagging and routing with maps, and the efficiency with tables. It answers research question Q3.

Chapter 6 analyzes the results and gives a conclusion about them. The answers to research questions are discussed and future work is proposed.

CHAPTER 2

Background

Commercial solutions to spatial problems may not always exist, or be accessible for organizational, financial or other reasons. Volunteered Geographical Information (VGI), which is edited by individual users, can fill the gap in the availability of digital geographical information (Goodchild, 2007). OpenStreetMap (OSM) is the most renowned actor of VGI, as it is a free editable spatial database of the world. OSM data can also be downloaded and reused with an open license (OpenStreetMap Wiki, 2022a).

This openness, in both data and specification, helped with the development of a whole range of applications around it, creating a sort of "ecosystem". The ecosystem of OSM applications provides a starting point for my solution, and paves the way for the development of derived work on an already established base.

OSM data is structured as follows. Elements can be separated into three types: nodes, ways and relations. On top of that, every type of element has thematic information stored in tags (pairs of key and value), as well as a unique identifier in their element type.

Nodes are the only elements that bear spatial information, as they have an associated latitude and longitude.

Ways are composed from nodes in a sequence, and can be open (start and end node are different) or closed (start and end node are the same). Closed ways do not necessarily reflect an area, as it depends on their tagging (thematical component). For example, a roundabout (tags: `highway=*` + `junction=roundabout`) is a closed way which does not constitute an area, but grass (`landuse=grass`) does.

Relations are composed of members, which are other elements of any type (nodes, ways or relations). These are used to represent more complex realities (e.g. public transport routes, boundaries, restrictions). For this project, the relation type that is relevant is multipolygon. Multipolygons (`type=multipolygon`) describe areas. Multypolygons can represent simple areas, such those that can be represented with closed ways, or more complex ones that cannot be represented with closed ways, such as polygons with holes or polygons with disjoint subareas (OpenStreetMap Wiki, 2022e).

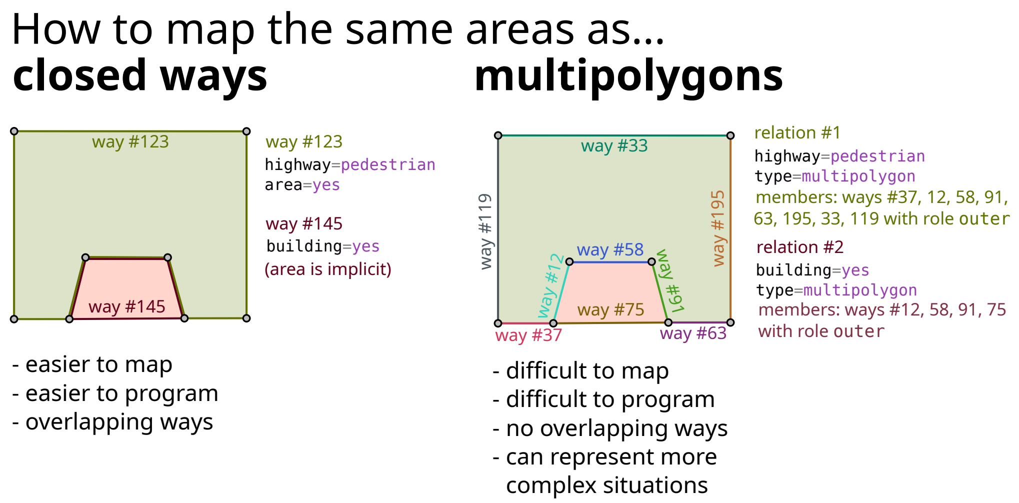


Figure 2.1: Comparison between closed ways areas and multipolygons for representing the same simple areas

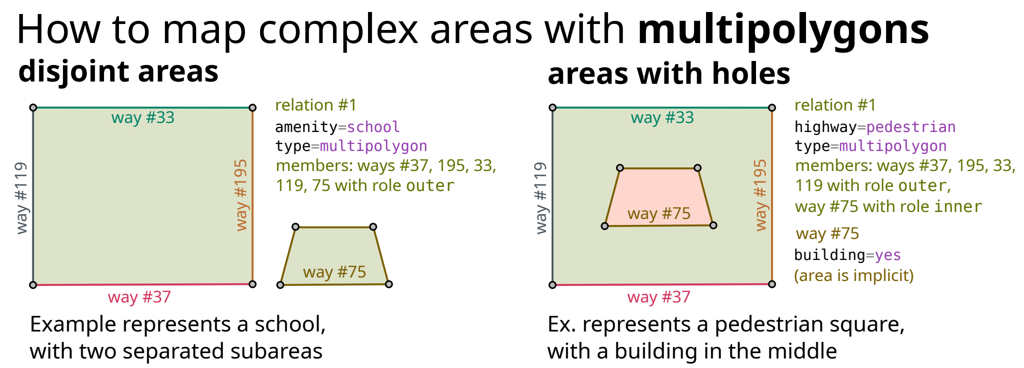


Figure 2.2: Examples of complex areas, mapped with the help of multipolygons

In the case of TDOAs, as areas are of utmost importance, closed ways and multipolygons are studied as part of the research.

The different mapping and tagging schemas are agreed on by the community, and explained at the OpenStreetMap Wiki (OpenStreetMap Wiki, 2022d).

This thesis aims to develop a standard way (schema) of mapping TDOAs, deriving from the current OSM time dependent (OpenStreetMap Wiki, 2022c) and pedestrian open areas (OpenStreetMap Wiki, 2022b, 2022f) schemas.

In the OSM "ecosystem", we can find that OsmAnd (juhanjuku, 2015) and OpenRouteService (ORS) (Bundesministerium für Verkehr und digitale Infrastruktur, 2019; Rylov et al., 2022) are the only routing services, so far, able to understand up to some degree of the TD routing schema. Due to the mobile-only nature of OsmAnd, and the fact that ORS started the implementation for OA routing, makes the latter the preferred choice for this thesis. ORS is furthermore an extended version of Graphhopper, a well used and documented routing service.

After defining the schema, I will combine time dependent (Bundesministerium für Verkehr und digitale Infrastruktur, 2019; Rylov et al., 2022) and open area (Hahmann et al., 2018) routing algorithms to create a single unified routing service for TDOAs. The unified routing service uses OSM data, and can understand the new schema, to generate the routing graph.

A web client is also being developed from previous work (Butzer et al., 2021). It acts as a graphical interface to interact with the unified routing service. This web client has the options to set a start and end points, and a time. It also incorporates a map to visualize the route.

Both ORS routing service (Rylov et al., 2022) and web client (Butzer et al., 2021) are released with open licenses, allowing for reutilization of the code.

The tagging schema and the unified routing service are set to be proven in a real world example. The surveyed data is then incorporated and merged into the existing OSM database, and therefore openly shared for other uses. After that, the adequateness and effectiveness of the results obtained are analyzed, comparing the unified routing service with a routing service that cannot handle TDOAs. The analysis focuses on measurable variables like computing time for generating the routing graph and routes, needed memory for the graph and distance variability with the most efficient route. This follows the approach taken by Hahmann et al. (2018).

CHAPTER 3

Methodology

The workflow in this Master's thesis is explained in this chapter.

Firstly, I develop a standard way (schema) of mapping TDOAs, as shown in figure 3.1. After that, I program the routing service that can interpret the new schema. The schema and routing service are verified in real world examples, which need to be surveyed beforehand, and its effectiveness measured.

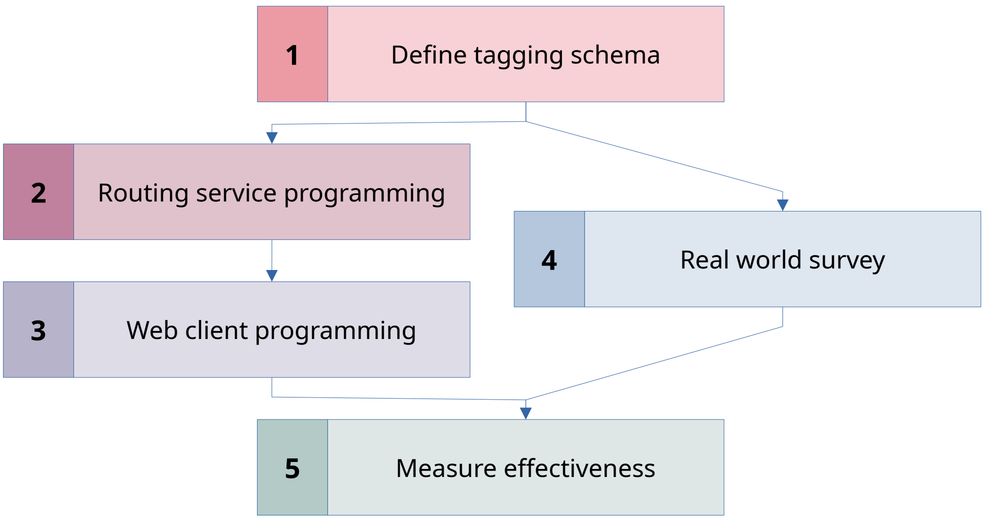


Figure 3.1: Workflow

3.1 Define tagging schema

A standard way (schema) of mapping TDOAs was developed, deriving from the current OSM time dependent (OpenStreetMap Wiki, 2022c) and pedestrian open areas (OpenStreetMap Wiki, 2022b, 2022f) schemas.

I have set up the following requirements for the tagging schema:

1. The schema must be futureproof, being broad enough so it can allow for multiple scenarios that may come up in the future. Therefore, I should avoid creating a schema that is too specific and can only work with few real world examples.

2. Older routing systems should still be able to route through the area/square/street that implements the new schema. Routing systems that do not implement open areas, time dependency or both combined must still return a route, even if it does not take into account all the detail given with TDOAs.

3. The schema must follow OSM philosophy: It should be adapted from existing schemas, making changes only when necessary, with little changes as possible, so the schema is well accepted by the OSM community.

The proposed tagging schema uses OSM areas (closed ways or multipolygons, depending on the case) to describe two different type of subareas: those that stay unchanged the whole time, which can be called "non-blocked", and those which change (or the access changes), which can be called "blocking".

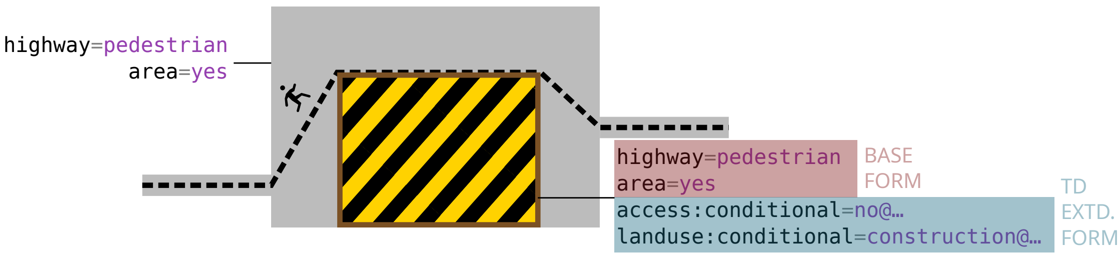


Figure 3.2: Proposed tagging schema, in use for a temporary construction site in the side of a square

Given a real world area that changes over time, "non-blocked" areas should have highway tags (pedestrian, living street...). Meanwhile, "blocking" subareas should have conditional access tags (OpenStreetMap Wiki, 2022c) in addition to the highway tags, plus any other information about why they are blocking (e.g. shop, construction, temporary building).

OSM data could be accessed in different times, as users could have the data downloaded (and kept offline) at a certain date. Because of this, data has to be made so changing features have thematic tagging that is correct for the most amount of time. Such areas should be mapped as not conditional (e.g. `highway=*`). This can be called the "base

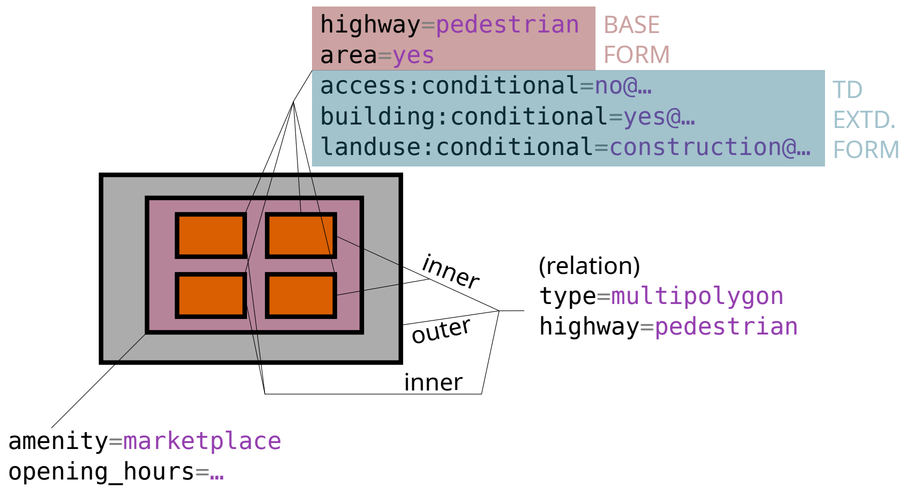


Figure 3.3: Proposed tagging schema, in use for a temporary market with stalls in the middle of a square

form" of the area. The thematic component that stays for the least amount of time should bear the conditional tag (e.g. `landuse:conditional=construction@...`). This can be called the "time dependent extended form" of the area. This ultimately ensures that routing systems which cannot handle time dependency can give an accurate routing for most of the time, as they will only be able to understand the "base form" of the area.

Regarding the two possible cases explained on chapter 1, figure 3.2 explains the proposed applied tagging for a temporary construction site on the side of a pedestrian square.

Figure 3.3 explains the proposed applied tagging for a temporary market in a pedestrian square, with the inner areas being the temporary stalls, having conditional access, building and landuse tagging.

3.2 Routing service programming

Changes to OpenRouteService (ORS) code (Rylov et al., 2022) were made, to adapt it to route through TDOAs.

As the original problem is too complex to achieve in one go, it was segmented into several subproblems, following the divide-and-conquer (*divide et impera*) approach. The subproblems and changes made to code are described in the following subsections.

Code changes were uploaded to a fork of the ORS repository on GitHub, and separated into feature branches, later requesting pull requests to the original repository. Integration of my work into an already well established repository ensures future reusability, and continuity of the project beyond the scope of this thesis.

The code for each subproblem is also designed to work on its own, if future use cases do not need to use the full extent of TDOA routing, they could still benefit from the extensions made to the original routing service code.

3.2.1 Add pedestrian support to ORS-time dependent (TARDUR)

The TARDUR project (Bundesministerium für Verkehr und digitale Infrastruktur, 2019; GIScienceHD, 2020) added time dependent routing, only for vehicles, to ORS. This code was integrated into the master branch of the ORS GitHub repository (Rylov et al., 2022).

TARDUR code encodes flags into the graph edges, that can get activated depending on the defined date and time while routing through it.

My first changes started from there, and consisted in extending the time dependent routing from vehicles to pedestrians. Changes to `FootFlagEncoder.java` were made, based on the bicycle vehicle flag encoder code, and adapting it to the pedestrian needs.

A pull request was created in the original ORS repository (Ochoa Ortiz, 2022c).

3.2.2 Add areas support to ORS (closed ways)

A version of the visibility algorithm, for closed ways open areas, has been implemented in ORS (GIScienceHD, 2017). It derives from Hahmann et al. (2018) and Graser (2016). The algorithm approach is explained in figure 3.4.

First, the tower nodes (where two or more ways meet, including the closed way which forms the open area) are selected. Afterwards, a local temporary graph is generated, connecting every node with all the other nodes in the closed way, as long as the connection edge falls completely within the area. Routes are then computed using Dijkstra between the tower nodes using the local graph. The final graph is composed from the computed routes plus the area perimeter.

The visibility algorithm helps create optimal routes in distance, between the tower nodes and, by extension, with the whole road network.

However, the ORS code was not working anymore due to a bug. After discovering it was not working, I posted the problem in the forum (Ochoa Ortiz, 2022b), which after got picked up (Oleś, 2022b) by the University of Heidelberg developers, and fixed (Oleś, 2022a).

3.2.3 Add multipolygon support to ORS

Originally, multipolygon support was planned, so complex areas could be described (see figure 2.2). However, after programming for one month and not achieving the desired result, I decided to withdraw it to allow for the thesis to be completed in time. The methodology followed in that month is, in any case, explained here.

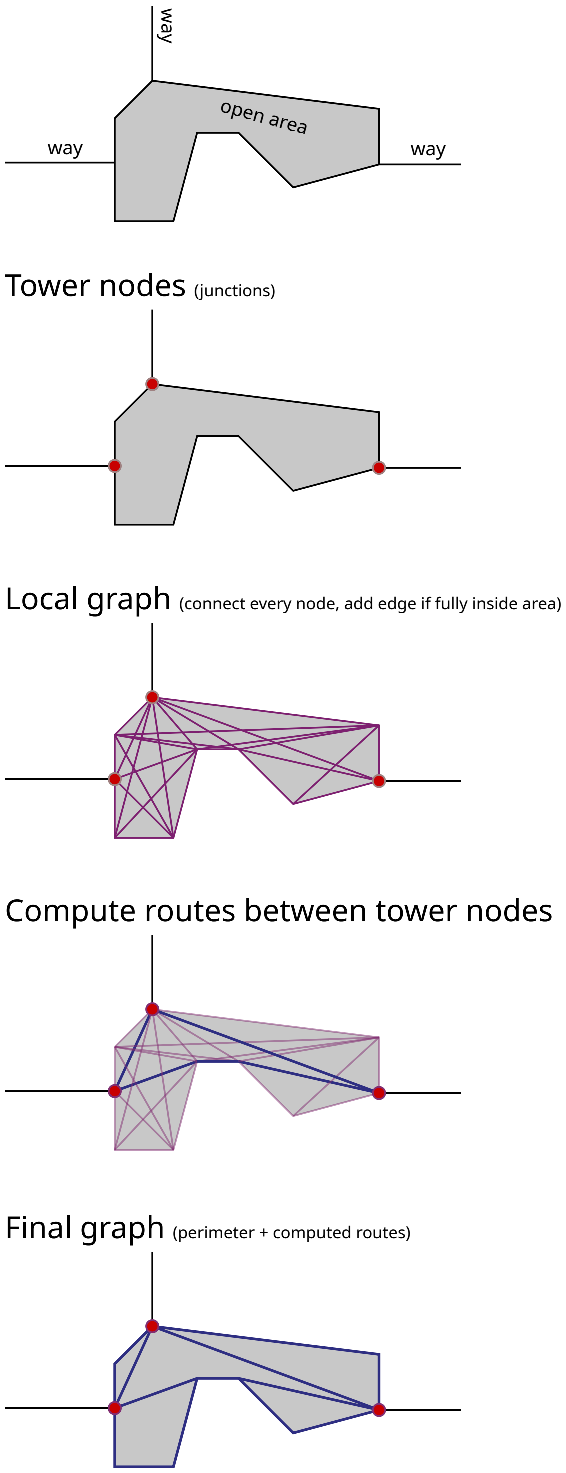


Figure 3.4: Open Area edge extraction algorithm from a closed way

ORS uses a forked Graphhopper library to read the OSM input files. This library goes through the file to store the necessary information to then turn nodes, ways and relations into edges on the graph.

The approach taken was to adapt the code that tackles turn relations. The ways involved in the relation are marked in preprocessing, so in processing they can be transformed into edges, with the correct edge flags.

However, the turn relations and the multipolygon relations differ in quite a lot. The ways that turn relations use already exist on the stored ways in preprocessing, as they are routeable because of its highway tags. On the other hand, multipolygon members, most of the time, do not have highway taggings. Instead the highway and therefore routing properties have to be inferred from the relation.

This meant that the whole data import approach from Graphhopper and ORS had to be redesigned, including which information gets stored for the ways and relations, and how eventual edge case scenarios can be tackled (e.g. two adjacent multipolygons, or highway way as member of multipolygon). This in turn meant also further redesigning of how time dependency and open areas are treated in ORS.

This was deemed ultimately out of scope and too complex as a subproblem inside of this project and is proposed instead as a standalone thesis in future work.

3.2.4 Combine the different subproblems

Combining subproblems 3.2.1 and 3.2.2 turned out to be trivial. As the code for both subproblems are in different steps of the graph generation, the edges from closed ways which form closed areas have the time dependency flags correctly applied, without further complication.

3.3 Web client programming

ORS TARDUR app (Butzer et al., 2021) was updated to be the web client for this project, removing the isochrone functionality and changing the routing from vehicles to only pedestrian.

Also, a bug concerning dates picked outside the current year was fixed (Ochoa Ortiz, 2022a).

Figure 3.5 shows the interface for the web client. It has a start and end point, departure/arrival time, a map and information about the current route.

The web client (frontend) connects to the routing service (backend) via its API, sending vehicle type (in this case pedestrian, no vehicle), coordinates, departure/arrival time and output file format. It then returns the information (in this case using the GeoJSON file format), that gets drawn in the map and its information (surface, elevation, way types, duration, distance) displayed.

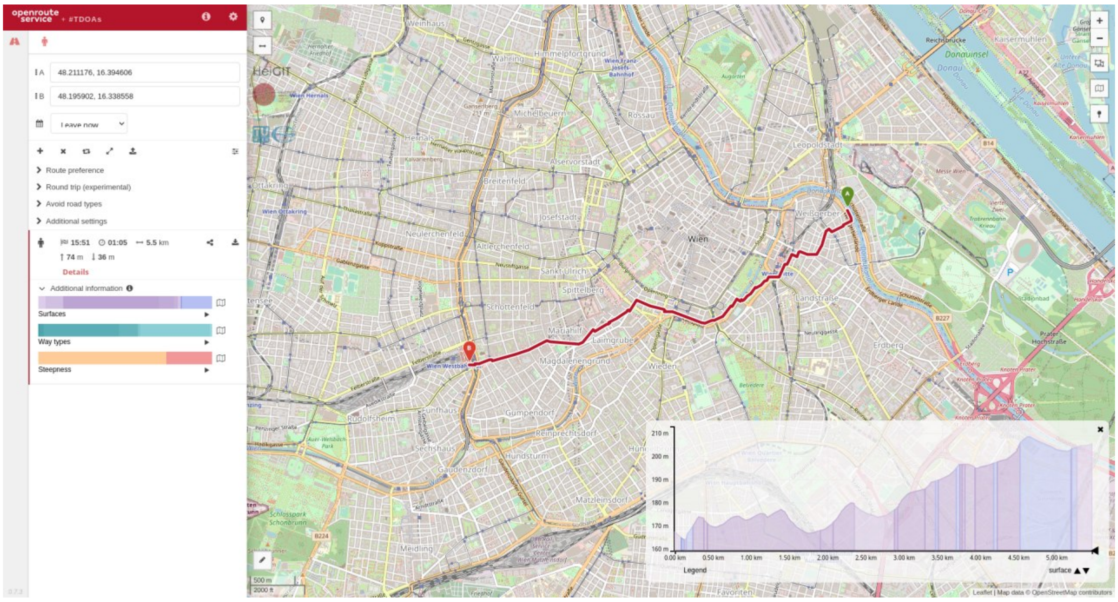


Figure 3.5: Web client interface

3.4 Real world survey

I surveyed with the Go Map!! phone application. This allows the user to edit directly the OSM database, with a background image. I used the Vienna city council "Mehrzweckkarte", combined with aerial imagery from the Austrian Government, and ground truth observations to add the data to the app.

Figure 3.6: Surveying at Altgasse, Hietzing

I did not use any speciallized tool for surveying, as precision is not the main focus in this survey. The focus is instead to showcase possible TDOA scenarios and how can they be represented in a database, routing applications and maps. City councils and other interested stakeholders could in the future pay surveyors to precisely represent changing areas (see chapter 6).

I made a shell script (listing 3.1) that generates an extract of the City of Vienna, with the latest available OSM data from GeoFabrik. The extract file is then loaded into the route service so the routing graph can be generated. Apart from OSM data, when processing, ORS fetches data from SRTM and GMTED for elevation (openrouteservice, 2022). This option could be removed for the small examples in this thesis, but was decided to be kept as the goal is to create a unified routing service for the whole city, that can handle TDOAs, rather than a specific TDOA routing service.

A diagram explaining the flow of data can be seen in figure 3.7.

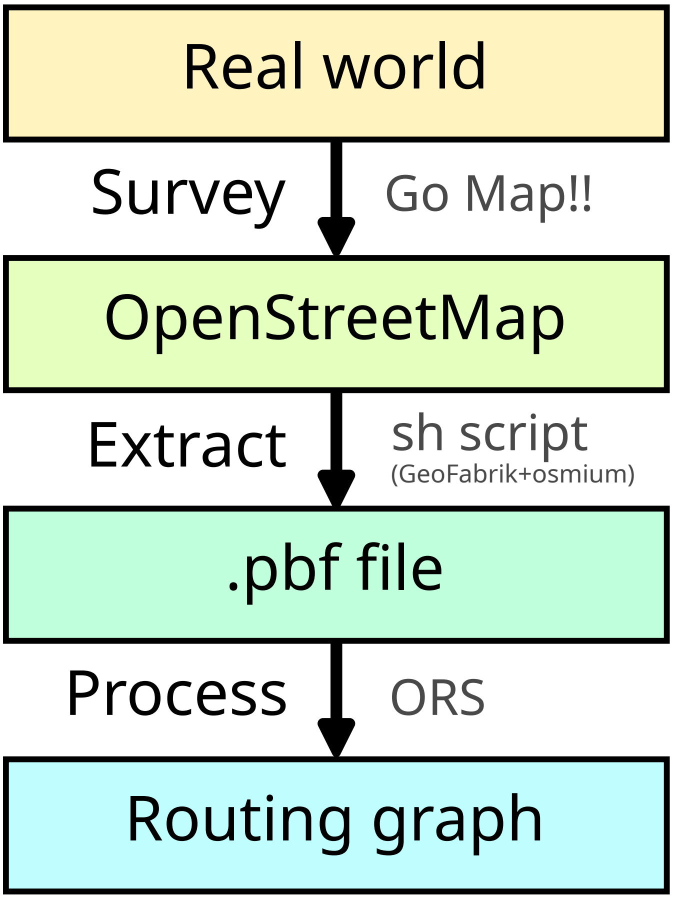


Figure 3.7: Data flow diagram, explaining the in/out formats or data states, the processes and the tools used

```
#!/bin/bash
# wien.sh: Downloads Austria latest PBF from GeoFabrik,
#          trims it into Vienna

# Download from GeoFabrik
wget https://download.geofabrik.de/europe/austria-latest.osm.pbf
# Export Vienna's boundary
osmium getid -r -t austria-latest.osm.pbf r109166 -o wien-boundary.osm
# Use boundary to trim
osmium extract -p wien-boundary.osm austria-latest.osm.pbf -o wien.pbf
```

Listing 3.1: Shell script for OSM data extraction

3.5 Measure effectiveness

An approach following (Hahmann et al., 2018) was taken. After generating the input data from the surveyed area, the backend (routing service) is started. The backend is set up with the necessary configuration for each analysis: without TD nor OA, with TD, with OA, or with both (TDOA).

After the graph is generated, the values of graph size and graph generation time are shown on the terminal. The time is calculated by storing the value of the computer clock at the start and end of the generation, and then computing the difference.

For the route, a start and end point that allow for a great visualization of the difference between approaches is chosen. The coordinates are passed to the backend API from the web client (frontend). A GeoJSON is returned as the route, and displayed on the client. The client also shows the route length.

CHAPTER 4

Case study

One TD, one OA and three (later two) TDOAs have been identified to demonstrate the routing ability of our product. This are Schottenring U-Bahn station, Rathausplatz, Altgasse (Hietzing), Karmelitermarkt's Bauernmarkt and Mariahilferstraße (Neubaugasse U2 construction).

4.1 Schottenring U-Bahn station

I tested the accurateness of the TD routing code in the Schottenring U-Bahn station northern access. This is a good location for such goal, as the station serves as a shortcut between Herminengasse and Promenadenweg (Donaukanal), avoiding having to go through a set of stairs outside the station.

(a) Herminengasse entrance

(b) Promenadenweg (Donaukanal) entrance

Figure 4.1: Schottenring U-Bahn station northern access

This route includes only a small area of the station, which is before the fare requirement point. Therefore, no valid ticket is needed to go through it. The route is also adapted to

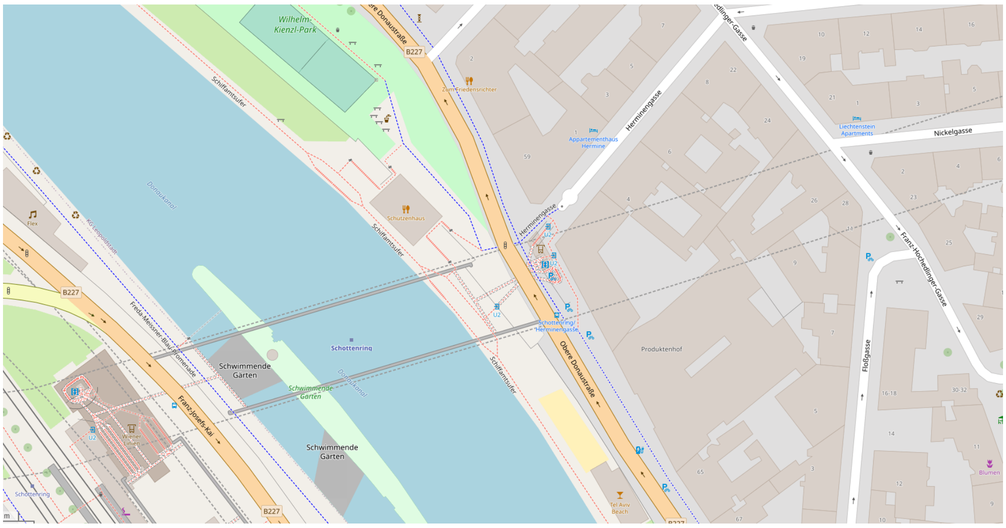


Figure 4.2: The northern entrance is located on the left bank of Donaukanal, in the corner of Obere Donaustraße and Herminengasse, 2nd district. Map data: OpenStreetMap contributors

people with reduced mobility, with the help of elevators. During the short time I was there, several people utilized the shortcut, realizing the great potential this one offers.

The station is open during train service times, and closed otherwise: from Monday to Friday from 0:30 to 4:45.

4.2 Rathausplatz

I tested the accurateness of the open area code at Rathausplatz. In Vienna it is difficult to find open areas mapped as closed ways in OSM. That is because it is typical for squares to have a statue or fountain in the middle, therefore needing OSM multipolygons, as holes are present. Rathausplatz can be represented as a closed way, as it has no permanent element in the middle.

Rathausplatz could be also considered as a TDOA example, as it often has temporary elements such as a stage and stalls for the Summer Film Festival, or an ice skate ring in winter. However, that would require very complex and detailed mapping, and survey all over a year, and also conversion to multipolygon to represent the complex reality. That is out of scope for this project, but it can serve as the perfect example for TDOAs in a future where the city council could get involved in the project, adding its detailed measurements and data to represent the different events available at the square over the year.

Figure 4.3: Rathausplatz is located between the Ring (Universitätsring) and the Vienna City Council building (Rathaus), 1st district. Map data: OpenStreetMap contributors

Figure 4.4: The Hietzing weekly market is located in the Altgasse, 13th district. Map data: OpenStreetMap contributors

4.3 Altgasse, Hietzing

A little unnamed square, at the Altgasse in Hietzing, is the location of a weekly farmers' market (Stadt Wien, n.d. b). This market takes place on Saturdays, from 9 to 16 (albeit the area is closed for traffic from 8 to 17, for market preparation time).

This area was chosen as the time is clearly signposted, and is small, which makes surveying and mapping easier. However, this location also poses an edge case, where an area usually devoted to vehicles becomes pedestrianized. This adds more complexity in the access tagging.

Figure 4.5: Altgasse temporary market. The sign on the right explains some temporary restrictions about the market.

4.4 Karmelitermarkt's Bauernmarkt

The Karmelitermarkt is one of the oldest still standing markets in Vienna (Wiki, 2021). It has 80 permanents stands, with a square that on Fridays and Saturdays host a temporary farmers' market, with 39 extra stands Stadt Wien (n.d. a).

Figure 4.6: The temporary farmers' market is located at the open pedestrian area, at the back of the static Karmelitermarkt. The market as a whole is between Im Werd, Krummbaumgasse, Leopoldsgasse and Haidgasse, 2nd district (Leopoldstadt). Map data: OpenStreetMap contributors

This location was chosen as it is a clear example of a temporary market, in an otherwise pedestrian square. Even though I have surveyed and mapped the market with multipolygons, it cannot be used in the final efficiency evaluation since multipolygon routing could not be implemented.

The tagging can still be considered a result, and will therefore be explained in the results section.

Figure 4.7: Karmelitermarkt temporary farmers' market stalls

4.5 Mariahilferstraße (Neubaugasse U2 construction)

After Karmelitermarkt had to be dropped, I had to choose a secondary location for the analysis. I decided to also go for a pedestrian area that changed over time, in this case because of the construction of the U2 U-Bahn line extension, an area of Mariahilferstraße (in front of Mariahilferkirche) had to be closed temporarily.

This area will be a new entrance to the Neubaugasse U-Bahn station, for lines U2 and U3. The area is enclosed by fences, and access to shops on the even numbers side is still allowed through the open sidewalk.

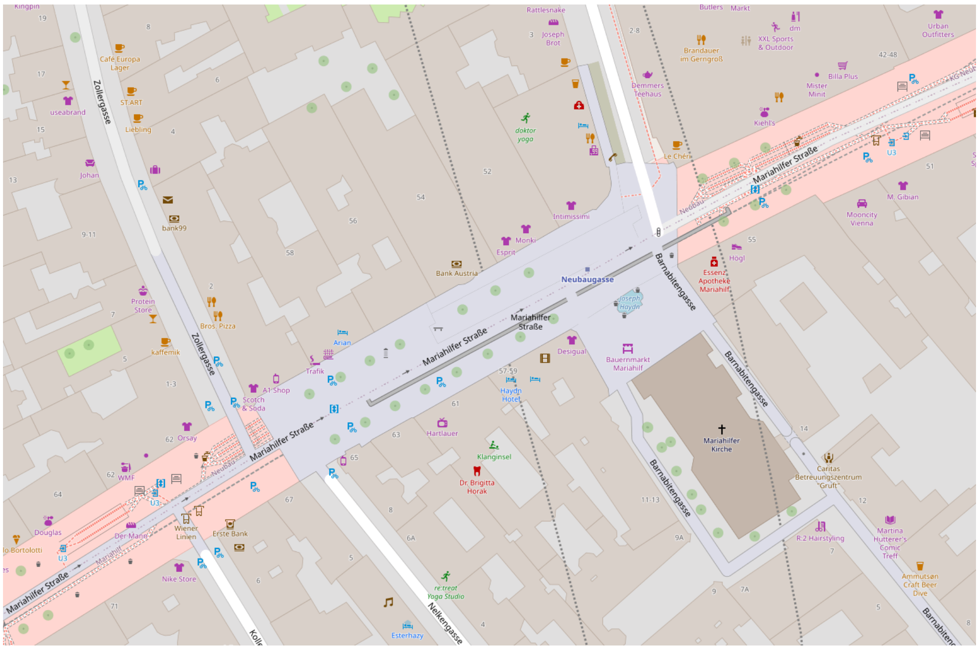


Figure 4.8: The construction site is located at Mariahilferstraße, in front of Mariahilferkirche

Figure 4.9: Photo of Mariahilferstraße case study. Explanation of the works can be seen on the fences.

CHAPTER 5

Routing results

I present case study routing results in this chapter, that demonstrate the efficiency of the routing service in each different real world example. To ensure results are comparable, they have been measured in the same machine with the specs shown in table 5.1, and with the graph generated from the same data extract.

For the data extraction, a script (see listing 3.1) has been programmed. The Austria extract from GeoFabrik is downloaded. Then with Osmium tool, the relation with id 109166 (Vienna municipality boundary) is exported. Afterwards, an extraction is made from the Austria file, using the Vienna boundary. The data extraction has been made on the 1st of September 2022, meaning the data is from 31st August at 21:00 CET (19:00 GMT) (Geofabrik GmbH, 2022).

Model	CPU	RAM	Swap	OS version
Lenovo 80XE	Intel Core i3-7100U @ 2.40GHz	4GB	4GB (file)	EndeavourOS 2022.08.05

Table 5.1: Computer specs

TD	OA	Graph generation time (s)	Graph size (MB)	Graph edges
X	X	45.996	23.03	195251
✓	X	46.958	27.03	195251
X	✓	71.134	24.03	220546
✓	✓	73.118	28.03	220546

Table 5.2: Comparison of graph generation time and graph size with the different options activated or deactivated

Comparison is made in table 5.2 between the graph computation time, graph size and graph edge number.

As seen in that table, adding TD does not increase the graph generation time in a great amount, but does increase the graph size by a considerable amount, while mantaining the graph edge number. This can be explained as TD is implemented by adding extra bits (flags) to each edge in the graph, being that a quick operation in computation time.

On the other hand, adding OA increases considerably the graph generation time and number of created edges, with the graph size increasing because of the number of edges, and not because of the amount of information in each edge. This is also because of how it was implemented (see figure 3.4), as no extra information is added to each edge, but rather extra edges are generated connecting the tower nodes with each other.

This extra computation can be compared with the number of elements with TD, OAs and TDOAs schemas applied. This information can be seen at table 5.3.

Elements	Number (% total)	Source
Total highway ways	**105254 (100%)**	https://overpass-turbo.eu/s/1lzq
Highway ways with TD schema	625 (0.6%)	https://overpass-turbo.eu/s/1lzk
Highway ways with OAs schema	1312 (1.25%)	https://overpass-turbo.eu/s/1lzo
Highway ways with TDOAs schema	28 (0.03%)	https://overpass-turbo.eu/s/1lzm
Total highway multipolygon relations (OAs)	**164 (100%)**	https://overpass-turbo.eu/s/1lzs
Highway multipolygon relations, with TD (TDOAs)	3 (1.83%)	https://overpass-turbo.eu/s/1lzu

Table 5.3: Ways statistics for OSM data inside Vienna city boundary, at the extraction time, fetched via Overpass Turbo

5.1 Schottenring U-Bahn station

Figure 5.1 gives an overview of the tagging of this case study in OSM. Highways relevant for routing are highlighted in orange, together with the U2 railway platforms under the canal in purple. In dashed yellow the ways with conditional (time dependent) tagging can be seen, this corresponds to the U-Bahn entry and the small mezzanine. Indoor ways have a gray semi-opaque overlay, being those inside the station, that connect all the way to the platforms. Not all the ways inside the station have conditional tagging, even if they could or should, but as the entrance and mezzanine areas (those that can be accessed legally without ticket) have conditional tagging, a routing service will never route through the remaining parts of the station anyway.

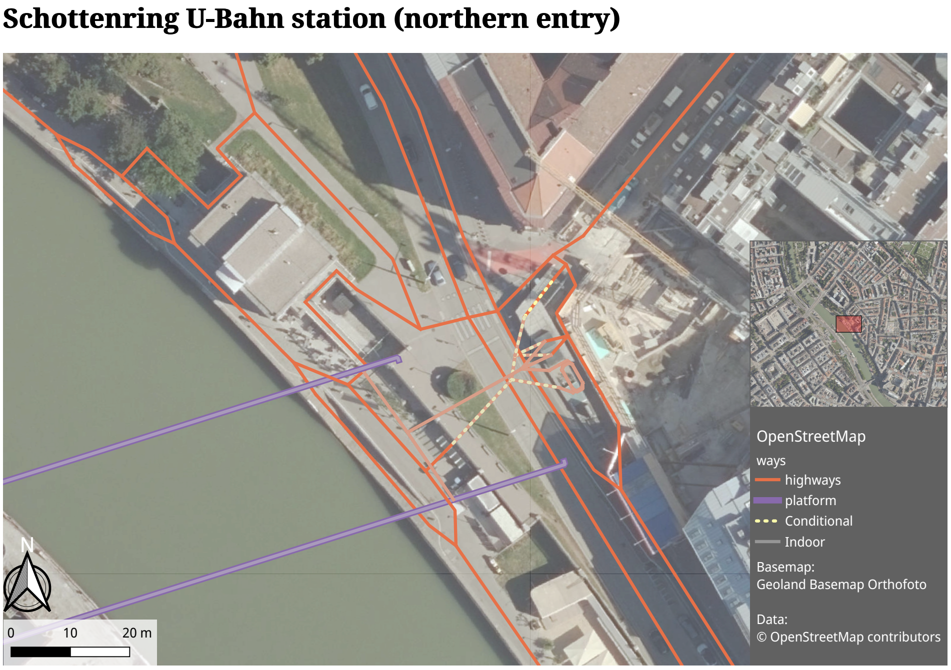


Figure 5.1: Highways relevant for tagging at the Schottenring case study

Time dependency is tested in this example. Comparison is made in table 5.4 for a route between the coordinates (48.21783, 16.37342) and (48.21733, 16.37308), with TD routing deactivated and activated in the service.

The figure 5.2 shows a visual comparison between routes when the station is open and closed. The route is a 62% longer when not going through the station. However, when not taking into account this time dependency, the routing system gives a route impossible to take when the station is closed (I marked it as ⚠on the table).

TD	OA	Dep. time	Route length (m, %opt.)
X	X	Not applicable	73m (optimal) ⚠
✓	X	Tue, Aug 30th 2022, 15:00	73m (optimal)
✓	X	Tue, Aug 30th 2022, 02:00	118m (+62%)

Table 5.4: Comparison of route length with the different options activated or deactivated, at the Schottenring case study

(a) Route when station is open

(b) Route when station is closed

Figure 5.2: Route comparison at Schottenring U-Bahn station northern access

5.2 Rathausplatz

Figure 5.3 gives an overview of the tagging of this case study in OSM. Highways (mapped as not closed way area) relevant for routing are highlighted in orange, with closed ways area highways in red. This closed ways area highways are the ones treated in OA routing, as seen in figure 3.4.

Open areas are tested in this example. Comparison is made in table 5.5 for a route between the coordinates (48.21150, 16.35855) and (48.21038, 16.36081), with OA routing deactivated and activated in the service.

Rathausplatz

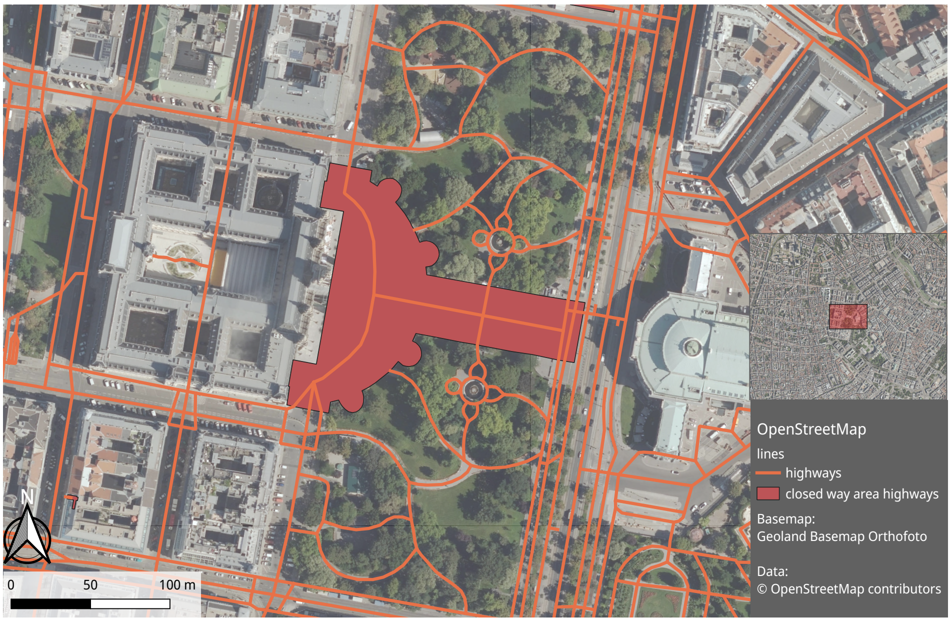
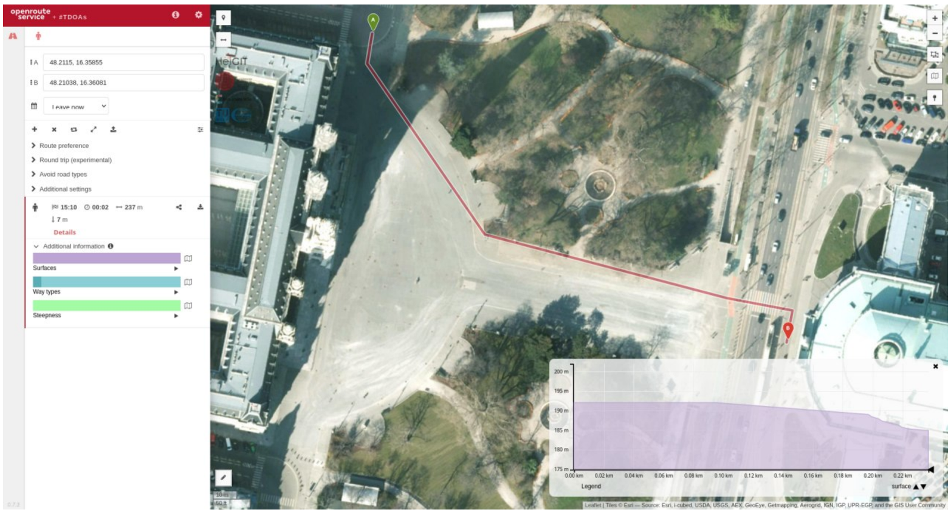


Figure 5.3: Highways relevant for tagging at the Rathausplatz case study

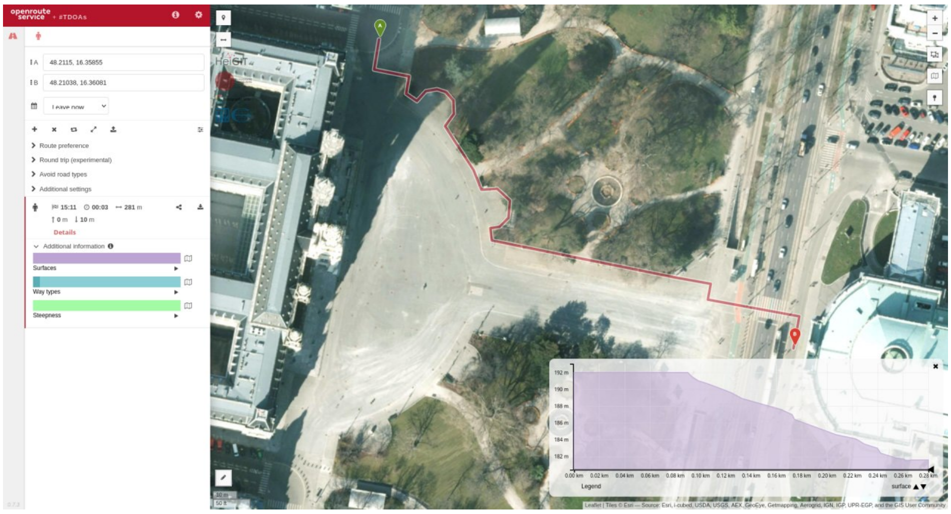
The figure 5.4 shows a visual comparison between routes when the option is activated or not. The route is a 19% longer when not activating the OA routing option, making a less efficient route. The route taken in that case follows the way perimeter, as the routing service is interpreting it as any other highway mapped as a way, and not understanding the open area.

TD	OA	Route length (m, %opt.)
X	X	281m (+19%)
X	✓	237m (optimal)

Table 5.5: Comparison of route length with the different options activated or deactivated, at the Rathausplatz case study



(a) Route when OA routing is active



(b) Route when OA routing is not active

Figure 5.4: Route comparison at Rathausplatz, when OAs are considered

5.3 Altgasse, Hietzing

Figure 5.5 gives an overview of the tagging of this case study in OSM. Highways (mapped as not closed way area) relevant for routing are highlighted in orange, with area highways in red. This closed ways area highways are the ones treated in OA routing, as seen in figure 3.4. Conditional access is seen with diagonal stripes, being vehicle conditional the striped-from-top-right-to-bottom-left area, and all (vehicle+pedestrian) conditional access the checkered board striped.

TDOAs are tested in this example. Comparison is made in table 5.6 for a route between

Altgasse, Hietzing

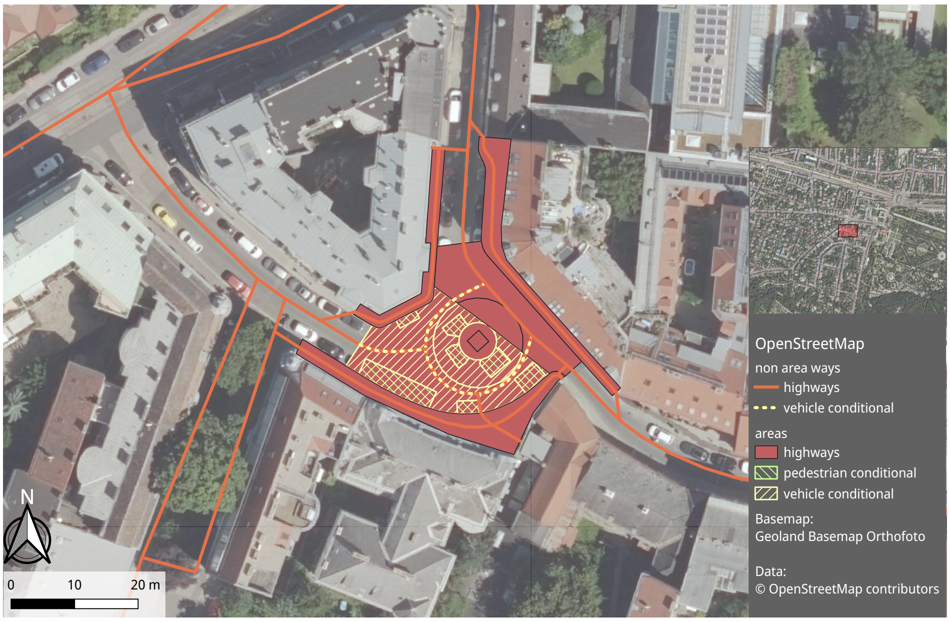


Figure 5.5: Highways relevant for tagging at the Altgasse, Hietzing case study

the coordinates (48.185608, 16.298164) and 48.185481, 16.298682), with TD andOA routing deactivated and activated in the service, and different departure times.

The figure 5.6 shows a visual comparison between routes depending on the routing options. The route is a 11% longer when not activating the TD and OA routing options, making a less efficient route. The route taken in that case follows the market perimeter, as the routing service is interpreting the areas as any other highway mapped as a way, and not understanding the open area.

The route with both options activated should be different depending on the day of the week. The market is open on Saturdays from 9 to 16, but when requesting a route at the market opening hours, the same route for pedestrians appears as when not in the market (⚠). This can be attributed to the adjacent area border problem, that will be explained in chapter 6.

TD	OA	Dep. time	Route length (m, %opt.)
X	X	Not applicable	49m (+11%)
✓	✓	Thu, Sep 1st 2022, 14:00	44m (optimal) ⚠
✓	✓	Sat, Sep 3rd 2022, 12:00	44m (optimal)

Table 5.6: Comparison of route length with the different options activated or deactivated, at the Altgasse, Hietzing case study

(a) Route without TD or OA routing active

(b) Route with TD and OA routing, outside of market hours

(c) Route with TD and OA routing, at market hours

Figure 5.6: Route comparison at Altgasse, Hietzing case study

5.4 Karmelitermarkt’s Bauernmarkt

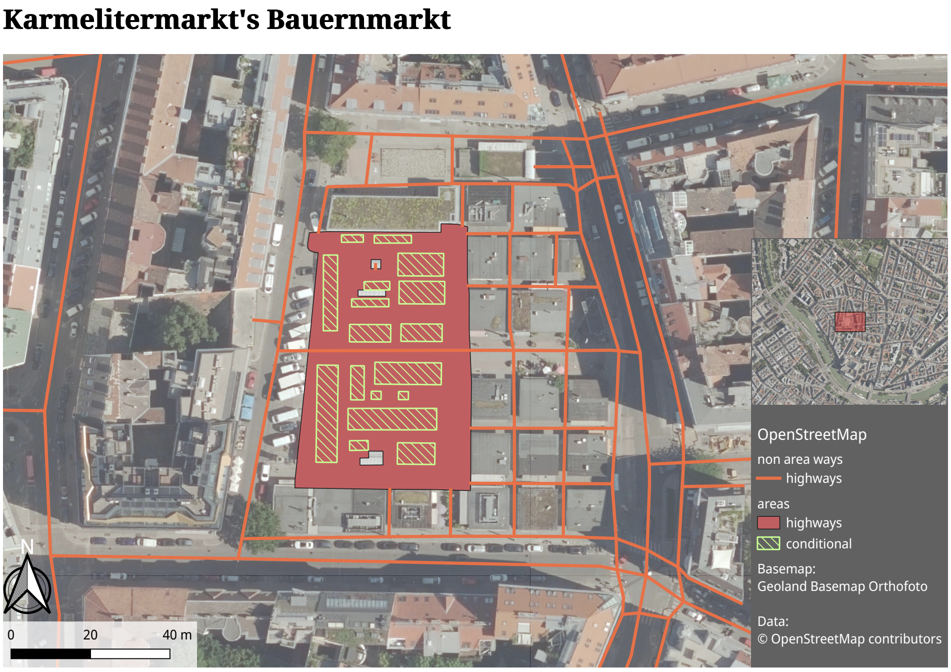


Figure 5.7: Highways relevant for tagging at the Karmelitermarkt case study

As seen in figure 5.7, the area was surveyed with multipolygons, and there was no way to resurvey it with closed way areas. This is because the stalls, together with the parking entrances, form holes in the square polygon.

It was decided to drop this case study for routing purposes, but it is still considered relevant for tagging and mapping.

5.5 Mariahilferstraße (Neubaugasse U2 construction)

Figure 5.8 gives an overview of the tagging of this case study in OSM. Highways (mapped as not closed way area) relevant for routing are highlighted in orange, with area highways in red. This closed ways area highways are the ones treated in OA routing, as seen in figure 3.4. Conditional access is seen with diagonal stripes.

TDOAs are tested in this example. Comparison is made in table 5.7 for a route between the coordinates (48.19912, 16.353155) and (48.199099, 16.351722), with TD andOA routing deactivated and activated in the service, and different departure times.

Mariahilferstraße (Neubaugasse U2 construction)

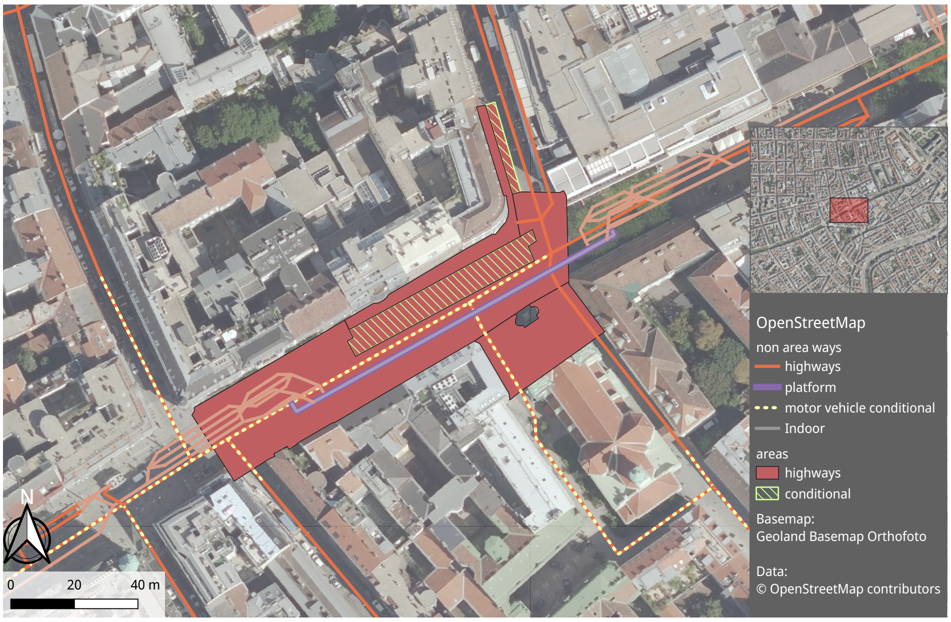


Figure 5.8: Highways relevant for tagging at the Mariahilferstraße case study

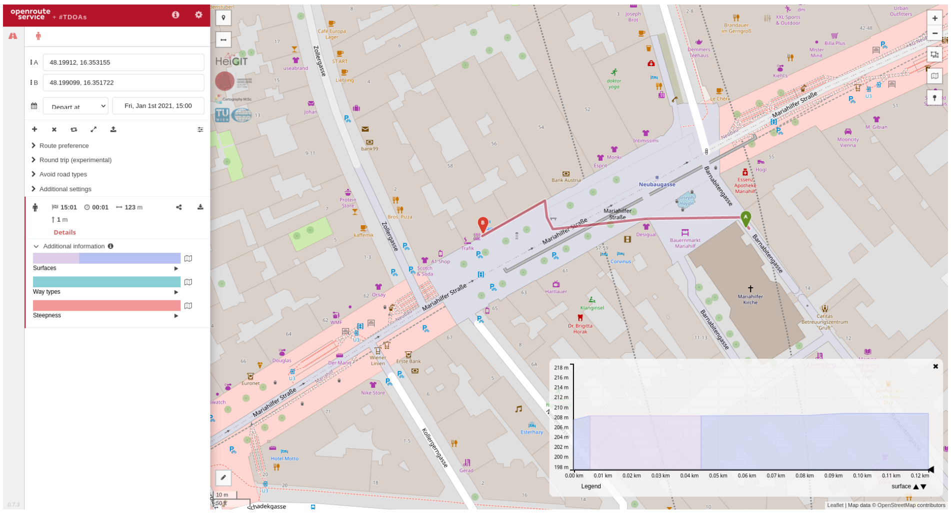
Figure 5.9 shows a visual comparison between routes depending on the routing options. The route is a 24% longer when not activating the TD and OA routing options, making a less efficient route. The route taken in that case follows the construction area perimeter, as the routing service is interpreting the areas as any other highway mapped as a way, and not understanding the open area.

The route with both options activated should be different at the days before the construction began. The construction began on the 1st March 2021, but when requesting a route before that date, the same route for pedestrians appears as when not in the market (⚠). This can be attributed to the adjacent area border problem, that will be explained in chapter 6.

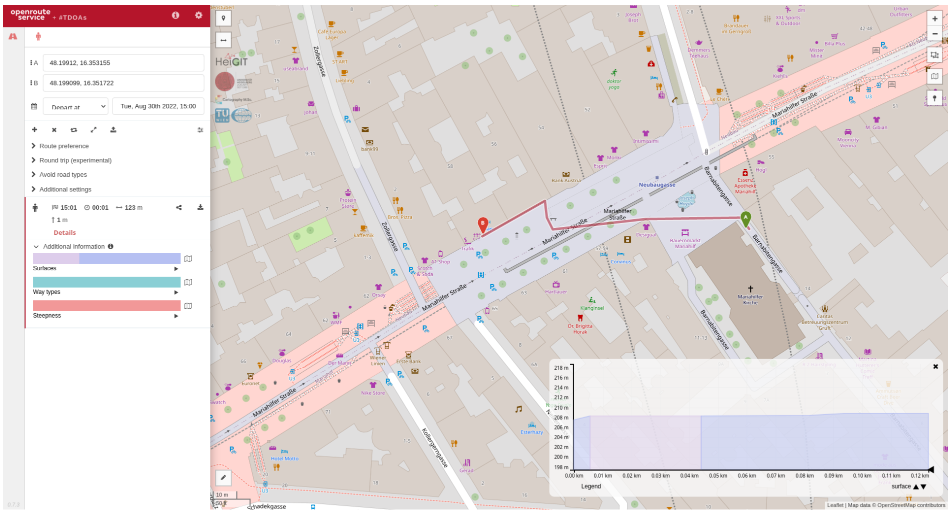
TD	OA	Dep. time	Route length (m, %opt.)
X	X	Not applicable	153m (+24%)
✓	✓	Fri, Jan 1st 2021, 15:00	123m (optimal) ⚠
✓	✓	Tue, Aug 30th 2022, 15:00	123m (optimal)

Table 5.7: Comparison of route length with the different options activated or deactivated, at the Mariahilferstraße case study

(a) Route without TD or OA routing active



(b) Route with TD and OA routing, before construction started



(c) Route with TD and OA routing, while construction is ongoing

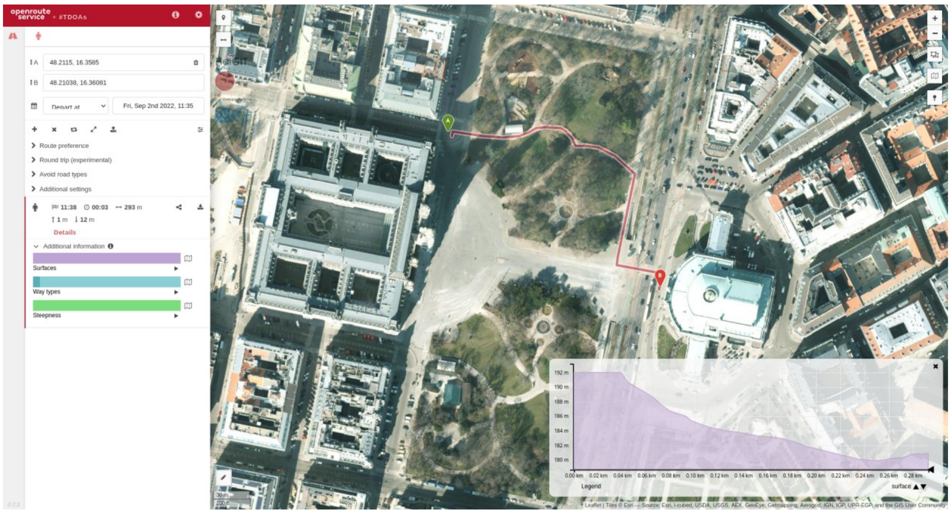
Figure 5.9: Route comparison at Mariahilferstraße case study

5.6 Extra results - Rathausplatz

To demonstrate that Altgasse and Mariahilferstraße results are due to the border effect, and not due to the program not understanding TDOA tagging, I created an extra dataset, modified after downloading from OSM.

I have downloaded Rathausplatz and its surroundings using JOSM, then added a simulated conditional access restriction to the square, from 8 to 17 hours, with the same program. This data is not uploaded to the live OSM database, but rather saved into a file, and then transformed into the input file with Osmium Tool for the routing service.

As seen in the figure 5.10, the routing service is able to correctly route through the square, or avoid it, depending on the time.



(a) Route with TD and OA routing, at simulated closing time

(b) Route with TD and OA routing, outside simulated closing time

Figure 5.10: Route comparison at Rathausplatz extra dataset

CHAPTER 6

Discussions and conclusions

In this thesis work, I introduce a new concept, TDOAs, that could be expanded and developed further, starting a new era of routing where pedestrian realities are more accurately portrayed. However, my work falls short because of not being able to include multipolygons at the end, and OSM limitations. Future work is expected in this realm to solve this pitfalls.

The use of OSM, while good for its easyness of entry and amount of tools built around it ("ecosystem"), it introduces some limitations for time dependency (e.g. in the real time domain). Data has to be the most time agnostic as possible, as users may use its data offline. On the other hand, not all the tools in the "ecosystem" are able to understand all the available schemas. That is why base and extended forms are introduced in tagging, so data is still valid for months to come, and tools can always understand the most simple base tagging even if they cannot handle the extended one.

TDOAs are demonstrated to be representable in OSM. My proposed schema fits the imposed requirements, and is able to adapt to a lot of realities before not mapped or considered, while still preserving OSM philosophy. The proposed schema was presented in section 3.1 and is the answer to the research question Q1.

I demonstrate in section 3.2 how to extend existing routing services, to understand pedestrian TDOAs. This is done following a divide-and-conquer (*divide et impera*) approach, separating the main problem into smaller subproblems. The routing service is also able to understand the routing service proposed in this thesis, answering research question Q2.

I explain the case study results in chapter 5. Great conclusions can be drawn from table 5.2. Adding TD does not increase the graph generation time in a great amount, but does increase the graph size by a considerable amount, while mantaining the graph edge number. Meanwhile, adding OA increases considerably the graph generation time and number of created edges, with the graph size increasing because of the number of edges, and not because of the amount of information in each edge.

This huge increase in computation leaves TDOA routing as an option that should be considered only for specific cases, where the better accuracy in routing is worth it.

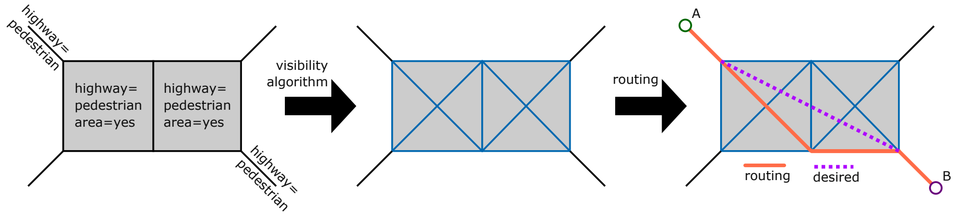


Figure 6.1: Adjacent area border problem

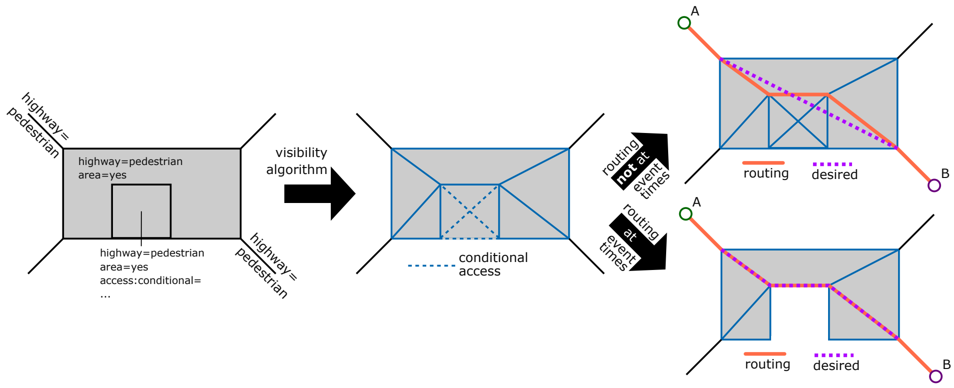


Figure 6.2: Adjacent area border problem, with TDOAs

Another problem the current solution poses is the usage of the visibility algorithm. Visibility algorithm is very efficient with single areas, but the "blocking" areas make a border problem arise. I call this "adjacent area border problem", and is explained in figures 6.1 and 6.2. As areas are separated between "non-blocked" and "blocking", the generated graphs from the "non-blocked" area will not go through the "blocking" one, rather going to the common nodes between both areas. In reality, at the times the "blocking" areas are still accessible, the efficient route would be to consider this areas as part of the "non-blocked". This effect can be clearly seen on both TDOAs case studies (tables 5.6 and 5.7). The border problem makes the routes similar or even the same length, when comparing times when conditional access flags are active or not active, and undermines the potential for TDOA routing in its current state.

Routing works correctly with TD or OAs alone, as long as there are no adjacent areas, and improves the fidelity to reality. This can be clearly seen in the respective case studies. Routing is proved to be possible for TDOAs, albeit the current approach is not perfect. Work should and must be redefined in future iterations. This answers research question Q3.

To sum up, several use cases and a lot of future work is proposed. This thesis should just be considered as the stepping stone in a much broader topic, yet to be developed.

6.1 Future work

In the future, multipolygon routing should be continued. This advancement should not only benefit TDOAs, but OSM routing as a whole. Area routing should not be different irrespective of their way of mapping (closed way or multipolygon), and mapping should reflect the reality on the ground, sometimes complex, not the data type most tools are able to understand. Tools have to be adapted to understand the reality.

Another interesting problem would be to fix the area border problem, either using new routing algorithms, or creating shared edges between adjacent areas (combining them when both are open, separated when not).

TDOAs could also be used by city councils or other bodies for modelling changes in their streets. For example, to portray access restrictions in construction sites. As seen in figure 6.3, currently the Vienna City Council just highlights in a map the streets affected, but their routing services are not altered.

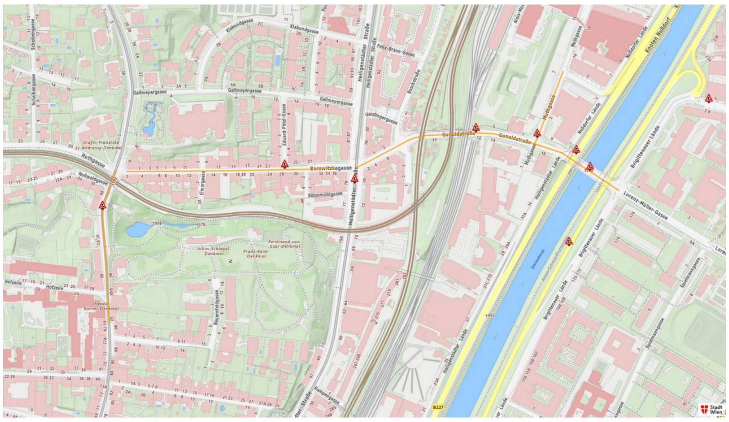


Figure 6.3: Streets currently in construction. Source: Vienna City Council

This example could be separated into two different classes: the restrictions that are already planned with a defined timespan (this could be described with the TDOAs approach taken in this thesis), and real time changes to areas (e.g. polygons that block access, fetched from an online feed). In the first one, the routing graph could be generated with the restrictions, as is done in this thesis. For the second one, changes should be made from my thesis approach. The real time application could be time critical if the time to recalculate the routing graph is deemed too large. Possible solutions to solve the problem would be to use a grid algorithm instead of a visibility algorithm, so paths can still be made in routing time around the polygons to avoid. Another solution would be to recreate the edges in a certain subgraph (of the area which routing is altered, instead of recreating the whole graph) then joining back with original graph.

A separate future problem, could be to better visualize the dynamic reality, including TDOAs, in maps. A tile server that handle conditional highways, landuses, buildings and access is proposed. It would be able to generate static (don't change over time) tiles that show the dynamic reality, or different versions for tiles that change over time. Then, integration of the tiles into the background layer of the routing web client.

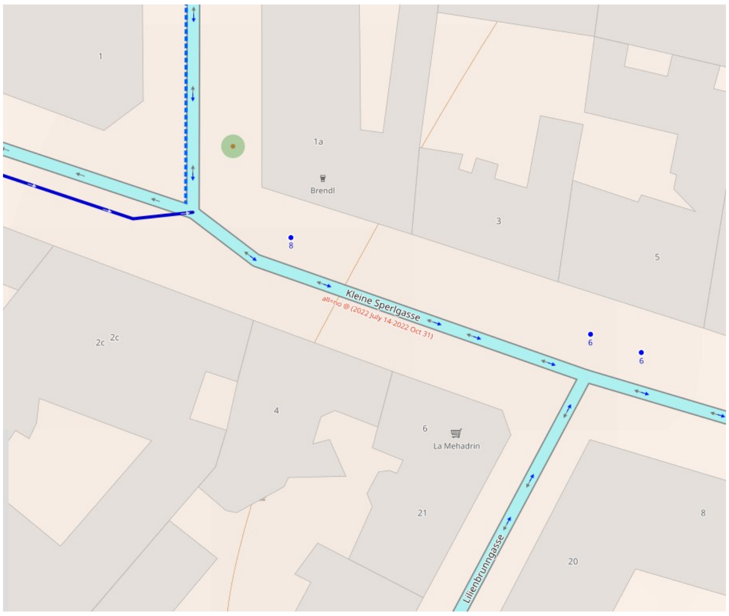


Figure 6.4: Time dependent tagging for ways shown on CyclOSM tiles. Source: cyclosm.org, OpenStreetMap contributors

As a last proposed future work, into the topic of pedestrian behaviour, how people navigate through TDOAs would be a main question to study.

List of Figures

List of Tables

Glossary

branch [software development] In software version control, a parallel life line of the code. A branch can later be merged into another branch, combining the changes made in both over time. 11, 12

flag [computing] An indicator inside a data structure, that can get turned on or off depending on context. 12, 14, 28

fork [software development] A separate copy of the original code, which then gets altered to achieve a distinct piece of software. 11, 14

pull request [software development] A request to pull or merge diverging code branches. It has to be approved by the developers of the repository/branch the merge will be made to. 11, 12

U-Bahn [transport] German speaking territories term for a metro (metropolitan railway). xxi, 19, 24, 28–30, 45, 46

Acronyms

API Application Programming Interface. 14, 17

CPU Central Processing Unit. 27

GeoJSON Geographic JSON. 14, 17

JOSM Java OSM. 38

JSON JavaScript Object Notation. 51

OA Open Area. 1, 2, 7, 17, 19, 27–37, 39, 41, 42, 46

ORS OpenRouteService. 6, 7, 11, 12, 14, 16

OS Operating System. 27

OSM OpenStreetMap. xv, xvii, xix, 1–3, 5–7, 10, 14–17, 20, 28, 30, 32, 35, 38, 41, 43, 44, 46, 47, 51

RAM Random-Access Memory. 27

TARDUR Temporal Access Restrictions for Dynamic Ultra-Flexible Routing. 12, 14

TD Time Dependent. 1, 2, 7, 17, 19, 27–29, 31, 33–37, 39, 41, 42

TDOA Time Dependent Open Area. xv, xvii, xix, 1–3, 6, 7, 9–12, 16, 17, 19, 20, 28, 32, 35, 38, 41–46

VGI Volunteered Geographical Information. 1, 5

Bibliography

Bundesministerium für Verkehr und digitale Infrastruktur. (2019). BMVI - Temporal Access Restrictions for Dynamic Ultra-Flexible Routing - TARDUR. Retrieved December 9, 2021, from https://www.bmvi.de/SharedDocs/DE/Artikel/DG/mfund-projekte/tardur.html

Butzer, A., Ellersiek, T., Nunes, R., Marquardt, M., Liu, L., C., A., Cudini, S., darkshadowlho, debyos, theonlytruth, Masson, Y., Chaput, A., Richter, K.-P., mircozorzo, not-a-kiwi, luisaGri, Rousell, A., Rylov, M., Lebedev, Y., … Raifer, M. (2021, June 30). *Classic openrouteservice maps client* [tardur tree]. Retrieved April 19, 2022, from https://github.com/GIScience/openrouteservice-app/tree/tardur

Geofabrik GmbH. (2022). Retrieved September 2, 2022, from https://download.geofabrik.de/technical.html

GIScienceHD. (2017). *GIScience news blog » blog archive » routing through open spaces – a performance comparison of algorithms*. Retrieved June 26, 2022, from http://k1z.blog.uni-heidelberg.de/2017/11/27/routing-through-open-spaces-%e2%80%93-a-performance-comparison-of-algorithms/

GIScienceHD. (2020). *GIScience news blog » blog archive » advanced time-dependent routing soon available in openrouteservice*. Retrieved August 24, 2022, from http://k1z.blog.uni-heidelberg.de/2020/04/28/advanced-time-dependent-routing-soon-available-in-openrouteservice/

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*, *69*(4), 211–221. https://doi.org/10.1007/s10708-007-9111-y

Graser, A. (2016). Integrating Open Spaces into OpenStreetMap Routing Graphs for Realistic Crossing Behaviour in Pedestrian Navigation [Publisher: Verlag der Österreichischen Akademie der Wissenschaften]. *GI_Forum 2016*, *4*, 217–230. https://doi.org/10.1553/giscience2016_01_s217

Hahmann, S., Miksch, J., Resch, B., Lauer, J., & Zipf, A. (2018). Routing through open spaces – a performance comparison of algorithms [Publisher: Taylor & Francis]. *Geo-spatial Information Science*, *21*(3), 247–256. https://doi.org/10.1080/10095020.2017.1399675

juhanjuku. (2015). *Maxspeed:conditional not working with date ranges · issue #1329 · osmandapp/OsmAnd* [GitHub]. Retrieved September 6, 2022, from https://github.com/osmandapp/OsmAnd/issues/1329

Ochoa Ortiz, H. (2022a). *Fix #368 by robot8a · pull request #369 · GIScience/openrouteservice-app* [GitHub]. Retrieved August 27, 2022, from https://github.com/GIScience/openrouteservice-app/pull/369
Ochoa Ortiz, H. (2022b). *Labs: Openrouteservice + open areas - openrouteservice - openrouteservice*. Retrieved August 3, 2022, from https://ask.openrouteservice.org/t/labs-openrouteservice-open-areas/3946
Ochoa Ortiz, H. (2022c). *Time dependent pedestrian routing by robot8a · pull request #1185 · GIScience/openrouteservice*. Retrieved August 15, 2022, from https://github.com/GIScience/openrouteservice/pull/1185
Oleś, A. K. (2022a). *Fix open spaces graph builder by aoles · pull request #1187 · GIScience/openrouteservice* [GitHub]. Retrieved August 3, 2022, from https://github.com/GIScience/openrouteservice/pull/1187
Oleś, A. K. (2022b). *Graph builder for open spaces fails to add new edges · issue #1186 · GIScience/openrouteservice* [GitHub]. Retrieved August 3, 2022, from https://github.com/GIScience/openrouteservice/issues/1186
openrouteservice. (2022). *Data - openrouteservice documentation*. Retrieved September 4, 2022, from https://giscience.github.io/openrouteservice/Data.html
OpenStreetMap Wiki. (2022a). *About OpenStreetMap - OpenStreetMap Wiki*. Retrieved April 4, 2022, from https://wiki.openstreetmap.org/wiki/About_OpenStreetMap
OpenStreetMap Wiki. (2022b). *Area - OpenStreetMap Wiki*. Retrieved April 18, 2022, from https://wiki.openstreetmap.org/wiki/Area
OpenStreetMap Wiki. (2022c). *Conditional restrictions - OpenStreetMap Wiki*. Retrieved April 18, 2022, from https://wiki.openstreetmap.org/wiki/Conditional_restrictions
OpenStreetMap Wiki. (2022d). *Map features - OpenStreetMap Wiki*. Retrieved August 20, 2022, from https://wiki.openstreetmap.org/wiki/Map_features
OpenStreetMap Wiki. (2022e). *Relation:multipolygon - OpenStreetMap Wiki*. Retrieved April 18, 2022, from https://wiki.openstreetmap.org/wiki/Relation:multipolygon
OpenStreetMap Wiki. (2022f). *Tag:highway=pedestrian - OpenStreetMap Wiki*. Retrieved April 18, 2022, from https://wiki.openstreetmap.org/wiki/Tag:highway%5C%3Dpedestrian
Rylov, M., Baumbach, T., Rousell, A., Leuschner, H., Oleś, A. K., Schnell, J., Marquardt, M., Psotta, J., Ellersiek, T., Kogios, A., Liu, L., Fendrich, S., Nolde, N., Butzer, A., Troilo, R., Bayerlein, C., Wang, Z., Higashi, S., Visintini, J., … Caldas, A. (2022, April 14). *Openrouteservice*. GIScience Research Group; HeiGIT. Retrieved April 19, 2022, from https://github.com/GIScience/openrouteservice
Stadt Wien. (n.d. a). *Karmelitermarkt, früher Markt im Werd - Öffnungszeiten, Lage*. Retrieved June 13, 2022, from https://www.wien.gv.at/freizeit/einkaufen/maerkte/lebensmittel/karmelitermarkt.html
Stadt Wien. (n.d. b). *Temporärer Markt Altgasse - Öffnungszeiten, Adresse, Angebot*. Retrieved June 13, 2022, from https://www.wien.gv.at/freizeit/einkaufen/maerkte/lebensmittel/altgasse.html

Wiki, W. G. (2021). *Karmelitermarkt — wien geschichte wiki.* Retrieved July 29, 2022, from https://www.geschichtewiki.wien.gv.at/index.php?title=Karmelitermarkt&oldid=800275