CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Computer Science

Master's Thesis

Generating context-enriched instructions for bicycle navigation

Petr Michalík
Study program: Open Informatics
Specialisation: Artificial Intelligence

May 2022
Supervisor: doc. Ing. Michal Jakob, Ph.D.

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Michalík Petr** Personal ID number: **474564**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence**

II. Master’s thesis details

Master’s thesis title in English:

Generating context-enriched instructions for bicycle navigation

Master’s thesis title in Czech:

Generování kontextem obohacených instrukcí pro cyklistickou navigaci

Guidelines:

Compared to car navigation, turn-by-turn cycling navigation presents a challenging problem. This is due to the complexity of the cycling environment consisting of various roads, paths and trails, as well as the limited ability of the cyclists to digest audio and visual information while outside on the bicycle. A possible direction for improving turn-by-turn cycling navigation is by generating more descriptive, context-enriched navigation instructions that would relate the route to its surroundings and, consequently, would allow the rider to better understand and hence more precisely execute required navigation maneuvers. The aim of this thesis is to explore the possibility for generating navigation itineraries consisting of such descriptive, context-enriched navigation instructions.

1) Survey existing approaches to descriptive, context-enriched navigation.
2) Propose a suitable algorithm for generating context-enriched navigation itineraries.
3) Implement the proposed algorithm.
4) Integrate the implemented algorithm with real-world map data from OpenStreetMap.
5) Evaluate the implemented algorithm on a representative set of test routes.

(For the avoidance of doubt: testing of the developed algorithm on users is not in the scope of the thesis.)

Bibliography / sources:

[1] Denis, Michel. "The description of routes: A cognitive approach to the production of spatial discourse." Current psychology of cognition 16 (1997): 409-458.

[2] Rousell, Adam, and Alexander Zipf. "Towards a landmark-based pedestrian navigation service using OSM data." ISPRS International Journal of Geo-Information 6, no. 3 (2017): 64.

[3] Dräger, Markus, and Alexander Koller. "Generation of landmark-based navigation instructions from open-source data." In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 757-766. 2012.

[4] Krukar, Jakub, Angela Schwering, and Vanessa Joy Anacta. "Landmark-based navigation in cognitive systems." KI-Künstliche Intelligenz 31, no. 2 (2017): 121-124.

Name and workplace of master’s thesis supervisor:

doc. Ing. Michal Jakob, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **28.01.2022** Deadline for master's thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

doc. Ing. Michal Jakob, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student's signature

Acknowledgement

I would like to thank my supervisors doc. Ing. Michal Jakob, Ph.D. and Ing. Pavol Žilecký for their guidance, support and insightful tips I received during the preparation of this thesis. I would also like to thank my family for their support throughout my study.

I also appreciate the contributions made by the volunteers of the OpenStreetMap project.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 20 May 2022

....

Abstrakt / Abstract

Lidé běžně popisují trasy s využitím orientačních bodů. V této práci navrhujeme metodu pro automatické obohacení navigačních instrukcí pomocí kontextových informací pro cyklistický navigační systém. Kontextem obohacená instrukce může odkazovat na orientační body (např. kostely, mosty, školy), tvar křižovatky, vlastnost cesty (např. cyklostezka, cesta vedoucí do kopce nebo z kopce), nebo na jinou výraznou vlastnost.

V této práci jsme rozdělili instrukce na tři typy podle jejich použití. Instrukce před křižovatkou slouží ke správnému rozpoznání místa odbočení, rozhodovací instrukce ke správnému odbočení v křižovatce a potvrzovací instrukce k ujištění uživatele, že jde po správné trase. Dále jsme popsali sadu operací pro začlenění všech instrukcí týkající se dané trasy do kompletního itineráře, které zahrnují odstranění nadbytečných instrukcí nebo popis složitých (cikcak) manévrů.

Naše řešení bylo demonstrováno pomocí webové aplikace, pomocí které lze nalézt a vizualizovat trasu společně s odpovídajícím itinerářem obohacených instrukcí. Výsledky naší práce byly vyhodnoceny na sadě testovacích tras, u kterých jsme ověřovali přítomnost předem zvolených orientačních bodů.

Klíčová slova: orientační bod, trasa, informace, cesta, význačnost, křižovatka

Překlad titulu: Generování kontextem obohacených instrukcí pro cyklistickou navigaci

Humans are used to describing routes using natural language with the use of landmarks. We propose a method for algorithmic enrichment of navigation instructions using context information for a bicycle navigation system. This context information can refer to landmarks (e.g., churches, bridges, schools), the shape of the junction, features of the road (e.g., cycleway, a path leading uphill or downhill), or any other property that stands out from the scene.

In this work, we have divided the instructions into three distinct types according to their use. The approach instruction is used to recognize the correct decision point, the decision instruction to find the correct continuation of the current route, and the confirmation instruction to assure the user that they are following the correct path. We then proposed a set of operations to merge all instructions related to the route into a single itinerary, which includes removing redundant instructions or descriptions of complex (zigzag) maneuvers.

Our solution was demonstrated in a Web application that can be used to find and visualize a route along with the corresponding itinerary of the enriched instructions. The performance of our work was evaluated on a set of test routes for which we checked the presence of predetermined landmarks.

Keywords: landmark, route, information, path, salience, junction

Contents /

Tables / Figures

Chapter 1
Introduction

1.1 Motivation

Most of the research on route planning has focused on planning the shortest or fastest route for cars. Although this approach is very well explored and works almost flawlessly for cars, it often fails to work for other means of transport — bikes and pedestrians. The main problems are the different available infrastructure and different granularity and structure of the instructions that best describe the route.

The fundamental difference is that pedestrians and cyclists have much greater freedom than cars. Cyclists can use cycleways, sidewalks, roads (except highways), traverse one-way streets, cross the road (not just at a zebra crossing), use shortcuts, use unmapped pathways, turn around and decide to ride back in the opposite way or get off a bike and behave like a pedestrian. We can already see that a cycling transport network is significantly larger and more complex than a car transport network.

As a result of this freedom, cyclists (or pedestrians) may need more information about their route, where each necessary detail can be provided using navigation instructions.

1.2 Aim of the thesis

In this work, we propose a technique for generating rich navigation instructions that can be utilized in navigation. The structure of the instructions is heavily inspired by the way people describe the routes in the natural language. These enriched instructions should be more comprehensible, sound more natural, and require less attention from the user. As a result, these instructions will help the user not to get lost throughout the route.

Following the route by visually checking the navigation device is not as problematic for a driver in a car as it is for a cyclist who needs to focus on the road and the environment. Cycling is physically more demanding in overall movement coordination and mentally demanding in keeping track of the surroundings. Therefore, we intend for these instructions to be given verbally (voice instructions), as we aim to eliminate the need to watch a navigation device along the entire route.

This work also focuses on generating an itinerary of enriched instructions with the help of a GPS device, where the route and itinerary are planned in advance. These context-enriched instructions can contain landmarks (e.g., churches, public transport stops, schools) or features of the route (e.g., cycleway, T junction, path leading uphill). Since the user needs different types of information, we have categorized the instructions into three types based on their use. These instructions are merged into an itinerary using a set of operations that transform and integrate these instructions together. In Chapter 5, we show details of our developed application that implements our approach and demonstrates the use of context-enriched instructions. The results of this work are shown and discussed in Chapter 6.

Chapter 2
Related Work

Creating a route and instructions describing this route so that the user successfully completes the route must incorporate knowledge from different fields of science: geography, psychology, linguistics, and computer science in the algorithmic generation of instructions. Early works in this area focused primarily on the linguistic and psychological part of the problem. However, with the spread of computers, mobile phones, and GPS devices, the last decade has witnessed a renewed interest in this field, mainly in the computer science part of the problem.

We have divided this chapter according to the way instructions can be generated. In Section 2.1, we describe instructions and itineraries generated by humans (such as when asking someone on the street for directions). When we plan a route between two points using a navigation system, these instructions are generated algorithmically. In Section 2.2, we describe the methods for generating context-enriched instructions and what type of context information these instructions should include. Section 2.3 then describes various sensory channels to communicate instructions to the user.

2.1 Human-generated instructions

Humans are used to describing routes using natural language with the use of landmarks and with minimal use of distance information. Therefore, this topic has been explored in a number of psychology works, in which the structure and function of navigational instructions have been studied.

Article [1] by M. Denis analyzed descriptions of routes to find how spatial knowledge is used in discourse. The author remarks that a large part of environmental knowledge is stored in the form of non-linguistic (presumably visuospatial) representations (often called “cognitive maps” or “mental maps”) and that this knowledge is being externalized when describing a route. Two significant components of the description of routes are references to landmarks and prescriptions of actions. Twenty undergraduate students created descriptions of a predefined route, and then those descriptions were analyzed. Each instruction was classified into one of five categories: actions without referring to any landmark, actions with landmarks, landmarks without referring to any action, description of landmarks, and commentary.

Each landmark was assigned one of three key functions: signaling sites where actions are to be accomplished (“Then you will see a church; go around it on the right”), locating other landmarks (“You will see a church; to the right of the church is a memorial; just to the right of the memorial is a path; take this path”), confirmation (“Walk about for 500 meters; you will pass a newspaper stand; then you will arrive at a crossroad where you will turn right”). Next, it was noted that most descriptions of routes could be accounted for by iteration of triplets of instructions: (re)orientate the person, start progress, announce landmark.

From all descriptions, a new skeletal description consisting only of key instructions was derived, which another group of participants selected. An interesting result of this part is that people can judge the relevance of route instructions, regardless of their knowledge of the environment of the route. From the skeletal description, it was concluded that landmarks and their associated actions are key components of the route description. This research also provided a characteristic of good descriptions. It is best to produce a limited number of statements, avoid redundancy, refer to visible, permanent, and relevant landmarks, and prefer determinate descriptions.

Article [5] studies the information requirements of pedestrians during navigation by conducting an empirical study with two groups of participants. In both groups, participants had to identify the information they felt that a pedestrian unfamiliar with the area would need to navigate those routes successfully. The resulting instructions frequently contained information on landmarks or the type of road. Additionally, pubs, supermarkets, and shops were often referred to by a specific name (brand). Also, 68% of all references were on decision nodes (intersections), and the rest were given along the path. This work also describes three purposes of information in instructions: preview ("turn left in 100 m when you see the pub"), identify ("turn left at Sainsbury's"), confirm ("turn left, the travel agency is then on your right-hand side"). These findings suggest that landmarks should be used as the primary means of providing directions to pedestrians, providing a name (where visible), pedestrian navigation should not rely on distance or road names, and confirmation instruction should also be used to maintain user confidence and trust. Another point of view is that street names are relatively stable compared to buildings that can change names or functions.

The next article [6] contradicts article [1] by M. Denis, in which it was stated that the only landmarks that are important for the quality of the route direction are those at the choice points and that the route directions rated as best are of moderate length. This work includes a more thorough exploration of the kinds and locations of landmarks included in the route directions, both at choice (decision) points and non-choice points. The results of the experiment with 31 students were that more than 50% of the landmarks on the unfamiliar routes and more than 40% of the landmarks on the familiar routes are mentioned at places other than the choice points. It was also shown that for both familiar and unfamiliar routes, longer route directions were rated as higher quality, and the longest segments were the most frequently mentioned (except for paths constrained by walls or bushes). This fact may be caused by giving the instructions in written form compared to being given orally in other studies. Another reason might be the complexity of the route since the path used in this work contained more turns and segments than in [1]. They also stated that the location of the landmarks in the plan seems to be related to the availability of the landmarks and the constraints of the route. The last finding is that for familiar routes, landmarks at potential choice points are used more often, as it is likely that one knows alternative routes (compared to unfamiliar routes).

Since it is not trivial to select an appropriate landmark for navigation instructions, the article [7] describes landmarks and how to measure their salience. The salience of landmarks can be attributed to visual attraction (area, shape, color, visibility), semantic attraction (cultural or historical importance), and structural attraction (intersections, squares, districts). In practice, some properties (e.g., size) are not part of the map data. Therefore, not all measures are applicable and valuable when working with real data.

This article is then extended by [8], which describes structural salience in more detail. As stated in this work, landmarks can be anything that stands out from the scene, acts as an anchor point for the user, and communicates the knowledge of the route. The shortcoming of article [7] is that it does not consider the advanced visibility (when approaching the decision point) and the position of the landmark relative to the route. Landmarks can occur off-route (distant or global landmarks), on-route at nodes, or on-route in between nodes. Next, the possible locations of the landmarks during reorientation (on decision nodes) and their effect are considered. Landmarks passed before the decision point are easier to conceptualize, as the maneuver occurs immediately after them, and they do not depend on the branching structure of the intersection ("Make a sharp right turn after the post office"). When the landmark occurs after the decision point, the instruction can be challenging to conceptualize and more complicated ("Make a sharp right turn at the intersection where the post office is at the opposite corner"). In addition, spatial chunking with landmarks is considered by changing the granularity of instructions for decision points without direction change. Therefore, the user can be instructed to pass the first landmark and execute a specified maneuver at the second landmark or to instruct the user to count the number of passed intersections or landmarks of the same type. When combined with other measures of salience, it was shown that this helps to select better landmarks for a route by considering the street network.

On the other hand, landmarks may not always be available or distinct enough. Article [9] analyzes the possibility of using street names as a substitute for landmarks. The advantage is that street names are displayed in a conventional fashion on street plates, freeing the user from any ambiguity. The drawbacks are that the name of the street might not be visible or known, they often do not convey any spatial information, are less effective, and are more cognitively demanding than landmarks.

Another contribution is article [10] that focuses on manually enriching a route with additional information. The main motivation of this work is that users of conventional navigation systems simply follow the instructions without actively mentally mapping the environment. The lack of environmental (survey) knowledge may lead to reactions to unexpected route deviations (traffic jams, road constructions). The data sources are Wikipedia (cultural, architectural, and historical objects) and Foursquare (popular tourist attractions) combined with the topographic data set of Germany. The result of this work is a set of enriched instructions for a selected route. Although these instructions are then compared with Google Maps instructions for the same route, no user experiments were conducted, so the influence of such instructions on people cannot be evaluated. Compared to other approaches, these instructions were significantly more wordy, which may give the user the impression that the system is more of a tourist guide than a navigation guide.

In summary, this section provides an insight into how people use landmarks for navigation and what people perceive as important during wayfinding. Although this work focuses on the algorithmic generation of instructions, these findings are useful for making generated instructions more human-friendly and understandable.

2.2 Algorithm-based instructions

Recent works regarding the routing instructions have focused on the computational part of the problem. Most of the articles have used the OpenStreetMap data as this is the most suitable and comprehensive dataset of map data from the whole world. Article [11] explores the possibility of including information about local landmarks in routing instructions. Landmarks were weighted according to their respective categories and scored based on their size, difference from the surroundings, proximity to the road, permanence, and other factors. The resulting online navigation system should generate routes with references to landmarks. Unfortunately, we were unable to check these results using the application demonstrated in this work. The proposed extensions by the authors are the selection of routes based on the ease of description, corrections ("If you reach the river, you have gone too far."), and chunking ("Turn left at the third intersection", mentioned in [12]).

The same approach was also chosen in article [13]. This article proposes five steps to generate instructions for a given route: splitting the route into episodes (between decision points), computing turn instructions (maneuvers), selecting landmarks at decision points using the suitability score, computing prepositions for landmarks (before, at, after), and generating instructions. The suitability score is determined by the distance from the decision point, category weight, relative location (the landmark located before the decision point is preferred), and the side of the landmark relative to the turn (the same side as the turn is desired). A similar approach can be seen in article [14]. The most salient landmark is selected for each navigational waypoint from a list of possible landmarks from the OpenStreetMap dataset. The suitability score is very similar to the suitability score in [13].

Another article using OpenStreetMap data for the extraction of landmarks is [2]. This work aims to develop pedestrian navigation, where each landmark is scored similarly to the previous articles. The selection of the most salient landmark depends on its category (tags in OSM), the distance from the reference point, which is a point before the decision point (in the approaching direction), visibility, location, and uniqueness. The authors also examine the coverage of landmarks in OpenStreetMap in different parts of the world, comparing the distribution of landmarks in the United Kingdom with Croatia. OpenStreetMap data in Croatia have slightly worse OSM completeness (in terms of landmarks), and the method is limited by the lack of landmarks.

This has been further extended in article [15] that generates landmark-based instructions. Compared to the other articles, the main difference is that this approach is not limited to selecting landmarks around decision points. The reason is that the instructions should be distributed at points where the class of road changes. Another addition is to consider administrative or environmental regions (parks) as landmarks.

The next article [3] focuses on enriching instructions for car navigation, recognizing that not only pedestrians can benefit from additional information about the route (landmarks). This work designed a system that uses OpenStreetMap data to plan a route, select landmarks complementing the driving maneuvers, and present it inside a driving simulator. The results of the experiments are fewer driving errors and fewer and shorter glances at the navigation display, which can minimize the risk of traffic accidents due to driver inattention.

Previous approaches have only considered creating each instruction independently and serving them directly as a list of instructions. These instructions can be automatically furthermore processed, as described in [12]. The core idea is to group several consecutive navigation actions, such as turns at intersections, into bigger units and use the different granularity of instructions for each part of the route. The instructions can be processed using numerical chunking ("Turn twice left"), chunking based on structural features ("Turn right at the T intersection"), based on linear features ("Follow the rail tracks"), and other types of operations with the instructions. These operations are then formulated as rewriting rules for a defined data structure using a choreme theory.

Previous works have only considered enriching navigational instructions for a predefined route. Thus, article [16] focuses on computing the simplest path, such that this path is easily and clearly describable, as opposed to the shortest path. When a path is computed, each pair of edges is given some weight, which reflects the amount of information required to negotiate the path between those two edges. The weighting function favors intersections, where the user just continues along the path or continues straight over a junction, and penalizes general intersections depending on their degree. The disadvantage of this algorithm is that the simplest paths are not symmetric, as the simplest path from A to B may be different from the route from B to A. Consequently, the simplest paths do not satisfy the triangle inequality due to this selected weighting function. The results show that the simplest path is, on average, only 16% longer than the shortest path, the algorithm does not use any distance measures, and the generated paths are more *cognitively plausible*.

An alternative approach to selecting landmarks can be seen in article [17], where data mining methods are used to discover landmarks in maps. This work applies the idea that a landmark can be anything that stands out from the scene. [8] Therefore, classification (supervised) or clustering (unsupervised) algorithms have been employed to determine which landmark is the most salient given its surroundings. This work used a wide range of attributes of buildings (use, size, distance, shape, the density of buildings around, orientation, parcel information, etc.) as inputs for the selected methods. The limitations of this work can be seen in using only buildings as landmarks and not evaluating the results with people.

Almost all of these studies have neglected the problem of timing to present route instructions. This perspective of the problem is described in [18]. The study participants walked through two routes, and the points in time and space at which the participants requested auditory route instructions were observed. The instructions were generated using the landmark-based method described in [2] and then manually altered due to poor salience or ambiguity. From the experiments, the authors have concluded that people unfamiliar with the environment want to receive instruction earlier on long segments, and thus these people might experience a higher degree of difficulty in wayfinding. From these results, we can see that people need some kind of confirmation that they are on the correct route when traveling on longer segments of a route.

All these ideas are summarized in journal [4], which brings a collection of articles related to this topic. All these articles include "the problems of identifying suitable candidates for landmarks in a scalable manner, selecting landmarks that are relevant to the navigating agent, communicating their presence on maps, and integrating them into location-based services". This journal also mentions the usage of landmarks in the augmented reality game "Pokemon GO" or "Ingress", where the landmarks were crowdsourced (manually picked by people), as opposed to being extracted automatically. The authors also mentioned that people differ widely in what they consider a useful landmark and that it is difficult to satisfy all possible use cases. There is also a difference in what is a cognitively plausible landmark from a technological point of view and a human perspective.

Next, article [19] in this journal asks the question of whether we *need* landmarks for navigation. In current car navigation services (finding the shortest path with GPS localization of the user), the landmarks are not needed. However, the use of landmarks may generate more efficient user-centered results, improve the understanding of the route visualization, and may communicate the instructions more efficiently. These landmarks can serve as meeting points or be used in a personalized navigation system, which are generated from the user's familiar landmarks.

Mitsubishi Electric Research Laboratories have developed a scene-aware interaction system [20–21] that translates sensing information into a natural language. One target application of this system is to provide drivers with intuitive route guidance. This system is capable of generating instructions such as “turn right before the postbox” or “follow that gray car turning right”. Furthermore, this system can provide voice warnings such as “a pedestrian is crossing the street” when this system detects an object that intersects the path of the car. All these features have been demonstrated in a video[1] with a car following a route and a system giving these more human-friendly instructions and warnings. The main difference is that this system relies on sensors and cameras to analyze the surrounding environment, whereas the approaches mentioned above have relied on map data. The advantage of this approach is that these instructions better reflect the real-world environment (map data may be obsolete). The disadvantage is that these instructions and the entire itinerary of the route cannot be prepared beforehand.

Since this thesis focuses on the algorithmic generation of context-enriched instructions, most of these articles will serve as a basis for our work and some (e.g., Mitsubishi scene-aware system) as inspiration.

2.3 Methods of instruction delivery

Another objective of including landmarks in route instructions is to minimize the amount of attention required during route finding. The primary sensory channel used during navigation is the visual channel (watching a display, reading a map, or reading written instructions). Navigation can also be achieved through the auditory or tactile channel, which is described in the following part.

In article [22] the authors have created a tactile belt with six vibrators (evenly distributed in all directions) that indicate directions and deviations from the path. Using this device should be unobtrusive since neither the visual field of the user nor the auditory sense is blocked. By not blocking these senses, users can avoid hazards such as traffic, obstacles, or other people. The evaluation of this system shows that people are able to perceive the direction of vibrations accurately and that this belt enables people to follow the route without any additional feedback.

2.3.1 Navigation of blind pedestrians

In articles [23–24], the authors have created a system for describing the environment to blind pedestrians using landmark-enhanced route itineraries. To implement this, a specially modified proprietary data model *PedestriNet* within the ROUTE4ALL project, which includes pedestrian links with higher positional accuracy, was developed. Navigation instructions are composed of an environment description to orient the user (street names, corners, crossings) and an action (direction, motion, slope, endpoint). Based on the results of the experiment with visually impaired participants, landmark-based navigation gave users an improved feeling of independence and was perceived as a safer and more effective method than metric-based navigation. The resulting product is accessible at `https://naviterier.cz/`.

An alternative approach to blind pedestrian navigation is navigation using teleassistance by another blind user, described in [25]. In this study, sessions where blind people were navigating each other remotely by phone were observed and analyzed. Consequently, a set of guidelines and successful navigation strategies have been extracted. The navigator should describe the environment in detail (e.g., paving blocks), inform

[1] `https://youtu.be/t0izXoT_Aoc`

the traveler about parked cars, check the position of the traveler regularly, the traveler should listen to the whole instruction before execution, etc. Furthermore, a POMDP-based dialogue system was created to replace the role of the human navigator with a computer system.

In this work, we intend the instructions to be announced verbally with the help of context information. The previous sections show that environment description is salient information that should be taken into account when enriching instructions with context information.

Chapter 3
Problem Specification

In this work, we focus on the algorithmic generation of instructions. Our approach for instruction generation allows for both taking a complete path, and optimizing the path based on the ease of description. From this path, an itinerary of instructions is created.

In this chapter, we first formalize the problem and the objective that we would like to achieve. Our critical goals are to ensure that the user does not deviate from the route and that the instructions provided are intuitively comprehensible. We define the terms graph, path, feature of vertices and edges, landmark, instruction, itinerary, and plan. Based on these terms, we describe our optimal itinerary and optimal route.

3.1 Landmark-enriched routing graph

For this task, we formalize the data as a directed graph. For each vertex and edge of the graph, we assign a point (set of points) in the Euclidean space $\mathbb{R}^2$, so that we can extract objects that are nearby.

3.1.1 Space

The majority of commonly available maps are stored in two dimensions, so we define our map object space as a Euclidean space $\mathbb{R}^2$. We use the distance function $d(x, y)$ between two points $x, y \in \mathbb{R}^2$, defined as $d(x, y) = ||x - y||_2$.

3.1.2 Routing graph

In this space, we use a routing graph in our space to build a structure that represents a road network.

Definition 3.1. A *directed graph* G is a tuple (V, E, f, g), where

- V is the set of vertices $v \in V$
- E is the set of oriented edges $e = (u, v), \quad u, v \in V$
  - u is a initial vertex (also called source or tail) of the edge e (denoted as $IV(e) = u$)
  - v is a final (also called target or head) vertex of the edge e (denoted as $FV(e) = v$)
  - self-loops are not allowed, i.e. $u \neq v$
  - parallel edges are not allowed, i.e.

$$\forall e_1, e_2 \colon (IV(e_1) = IV(e_2) \wedge FV(e_1) = FV(e_2)) \Rightarrow e_1 = e_2$$

- function g assigns a point in the Euclidean space $p \in \mathbb{R}^2$ to every vertex $v \in V$
- function h assigns a set of points in Euclidean space

$$h(e) = \{\, \alpha g(IV(e)) + (1 - \alpha) g(FV(e)) \mid \alpha \in [0, 1] \,\}$$

  to every edge $e \in E$

[26–28]

3.1.3 Neighborhood

Before we define a landmark, we need to represent the area surrounding each vertex and edge of our graph.

Definition 3.2. A *D-neighborhood* $N_D(v)$ of vertex $v \in V$ is a set of points

$$\{x \in \mathbb{R}^2 \mid d(x, g(v)) < D\}$$

and D is the size of the neighborhood.

Definition 3.3. A *D-neighborhood* $N_D(e)$ of edge $e \in E$ is a set of points

$$\{x \in \mathbb{R}^2 \mid d(x, y) < D, y \in h(e)\}$$

and D is the size of the neighborhood.

3.1.4 Features

Each vertex and edge in our graph have some features that can be communicated to the user via an instruction. These features can also be viewed as a type of landmark (something that stands out from the scene [8]), therefore we treat this type of information in the same way as landmarks.

Definition 3.4. The set F_e contains a set of all possible *features* (properties) of all edges (for example, the class of the road, the name of the road, or the elevation between the endpoints). [29]

Definition 3.5. The set F_v contains a set of all possible *features* (properties) of all nodes (e.g., the shape of the junction, the number of incoming and outgoing edges, or the type of the junction).

Definition 3.6. Function $f_e : E \to F_e^n$ assigns a subset of features $f_e(e_i) \subset F_e^n$ to the edge $e_i \in E$. [29]

Definition 3.7. Function $f_v : E \times E \to F_v^n$ assigns a subset of features $f_v(e_i, e_j) \subset F_v^n$ to the vertex $v_i = FV(e_i) = IV(e_j), \quad v_i \in V; e_i, e_j \in E$.

3.1.5 Landmark

Definition 3.8. A *landmark* l is a set of points in the Euclidean space $l \subset \mathbb{R}^2$. The set of all the landmarks $LM = \{l_1, l_2, \ldots, l_n\}$ forms our pool of candidate landmarks.

Based on this definition, a landmark can be a point, a line, or even a polygon. Since each vertex and edge are assigned a point (or multiple points) in the Euclidean space (using the function f or g), we can relate a landmark to each vertex.

Definition 3.9. A mapping $l_v : E \times E \to LM$ assigns landmark $l = l_v(e_i, e_j)$ such that $(l \cap N_D(v) \neq \emptyset) \wedge v = FV(e_i) = IV(e_j), \quad v \in V; e_i, e_j \in E$.

Definition 3.10. A mapping $l_e : E \to LM$ assigns landmark $l = l_e(e)$ such that $(l \cap N_D(e) \neq \emptyset)$ for edge $e \in E$.

3.2 Route

The result of our work is a route that consists of a path and an itinerary of instructions. This part defines these terms and describes how these terms are connected.

3.2.1 Path

To represent any path in our graph, we use the edges and vertices of our graph to define a path. Using this path, we can later create a route with the corresponding instructions.

Definition 3.11. A *path* W in the graph $G = (V, E)$ is a sequence of vertices and edges $W = (v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n)$, such that

- $v_i \in V$
- $e_i \in E$
- $e_i = (v_i, v_{i+1}), \qquad \forall e_i \colon v_i, v_{i+1} \in W.$
- $FV(e_i) = IV(e_{i+1}) \qquad \forall i \in [1, \ldots, n-1]$

In other words, a path is a sequence of vertices and edges. The vertices may repeat, or the first and last vertex may form a closed path.

3.2.2 Instruction

Methods in the literature (articles [2–3, 11–14]) have focused mainly on instructions that help the user identify the correct intersection. However, landmarks can also be used to choose the correct direction, keep the user on the determined path, and assure the user that they are following the correct path.

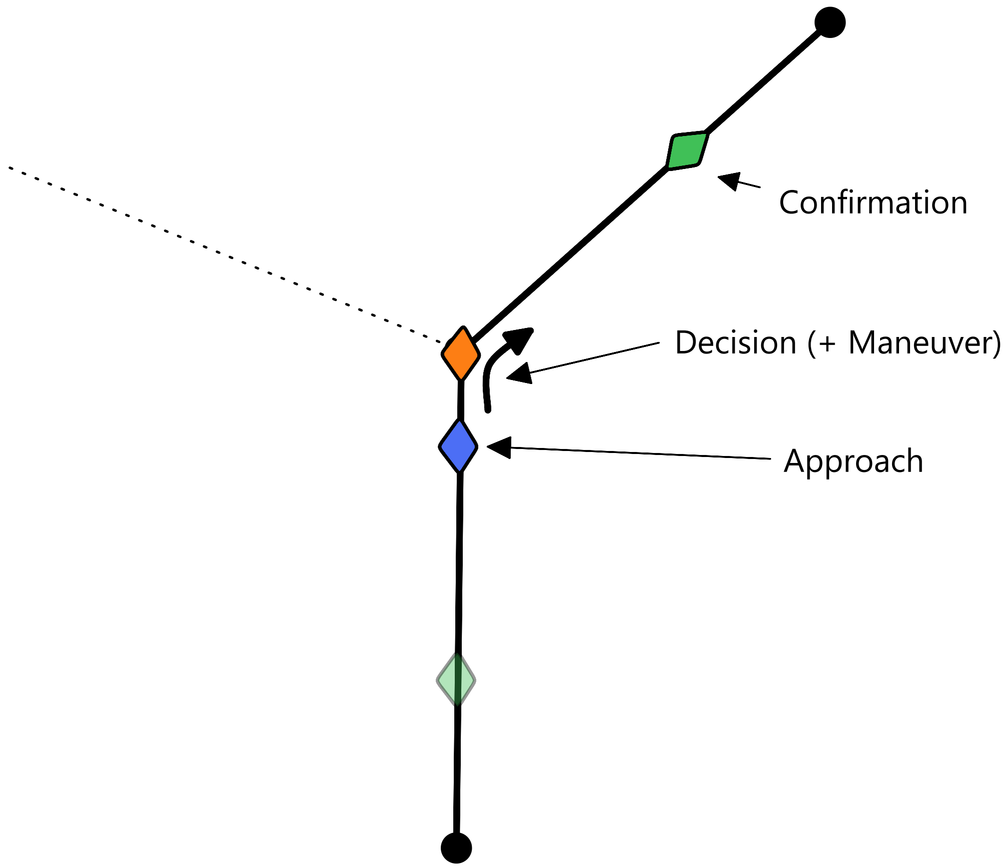


Figure 3.1. Types of navigation instructions

For this reason, we have split the instructions into three distinct types, each type containing a different piece of information: Approach instructions , Decision instructions, and Confirmation instructions. Maneuvers are regarded as information that can be part of the approach and decision instructions. By splitting the instructions into three types, we can keep the instruction short and concise, as each instruction provides only the piece of information that the user currently needs.

All these types of information are shown in Figure 3.1 in a simple Y junction, and the use case and content of these types of instruction are described later in Chapter 4 with details and use cases for each type.

Definition 3.12. A *decision instruction* at the intersection v of the edges e_i and e_j ($v = FV(e_i) = IV(e_j)$) is a tuple

$$I^D_{(e_i,e_j)} = (f_v(e_i,e_j), l_v(e_i,e_j), D^{(e_i,e_j)}_{start}, D^{(e_i,e_j)}_{processing}, m)$$

f_v are the features of the node, l_v is the associated landmark, $D^{v_i}_{start}$ is the distance in meters from the starting waypoint (distance along the path) to the vertex v_i, $D^{v_i}_{processing}$ is the distance in meters covered during the instruction announcement, and M is the related maneuver.

Definition 3.13. A *maneuver* m for a pair of consecutive edges e_i, e_j: $FV(e_i) = IV(e_j)$ is an item in the set of maneuvers

$$M = \{straight, slight\_left, slight\_right, left, right, sharp\_left, sharp\_right, back\}$$

This maneuver is a single piece of text information that is used when the user needs to be reoriented at a decision point.

The first instruction is an exception since there is no maneuver and no previous edge from which the user is coming.

Definition 3.14. The first decision instruction at the intersection v_1 and the edge e_1 is a tuple

$$I_{(v_1,e_1)} = (f_e(e_1), l_e(e_1), D^{(v_1,e_1)}_{start}, D^{(v_1,e_1)}_{processing})$$

Definition 3.15. An *approach instruction* at the edge e_i approaching the decision point v_k is a tuple

$$I_{(e_i,v_k)} = (f_v(e_i,e_j), l_v(e_i,e_j), D^{(e_i,v_k)}_{start}, D^{(e_i,v_k)}_{processing}, I_{(e_i,e_j)}, AD)$$

where AD is the distance from the oncoming decision point (with a decision instruction), $D^{(e_i,v_k)}_{start}$ and $D^{(e_i,v_k)}_{processing}$ have the same meaning as in Definition 3.12.

Definition 3.16. A *confirmation instruction* on the edge $e_k = (v_i, v_j)$ is a tuple

$$I_{e_k} = I_{(v_i,v_j)} = (f_e(e_k), l_e(e_k), D^{(v_i,v_j)}_{start}, D^{(v_i,v_j)}_{processing})$$

f_e are the features of the edge, l_e is the associated landmark, $D^{(v_i,v_j)}_{start}$ and $D^{(v_i,v_j)}_{processing}$ have the same meaning as in Definition 3.12.

3.2.3 Itinerary

Throughout the route, the types of instruction alternate. The first decision instruction is followed by a confirmation instruction, followed by an approach instruction that precedes a decision instruction. The route ends with the last approach instruction that announces the location where the route ends.

When we define an itinerary of instructions, not every instruction can be announced in time due to overlap with another instruction or due to redundancy. This selection method is described later (Section 3.3) and is one of the objectives of this work.

Definition 3.17. An *itinerary* $\mathbf{I}$ for a path $W = (v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n)$ is a sequence of instructions $\mathbf{I} = (I_{(v_1,e_1)}, I_{(v_1,v_2)}, I_{(e_1,v_2)}, I_{(e_1,e_2)}, \ldots, I_{e_{n-1},v_n})$, where $\mathbf{I}$ are the instructions as defined above and each item of the itinerary $\mathbf{I}$ is related to the path W based on its index.

3.2.4 Best continuation

The user should not be informed about every intersection that occurs on the route. When the route leads straight along the same (or similar) path, the user is most likely to continue correctly even without any instruction. Therefore, we formalize and describe the cases in which the user does not need an instruction to proceed correctly along the designated route. The *best continuation* is such an outgoing edge inside the intersection that the user will choose without any additional information about the required maneuver. The exact solution for selecting the best continuation is described later in Section 4.6. Generally, there are only two cases of the best continuation. Either the best continuation is not present (when there is no straight edge or there are multiple straight edges), or there is one outgoing edge that is marked as the best continuation of the current edge. An example with no best continuation can be seen in Figure 3.1 with a Y junction.

3.2.5 Decision point

Since we can recognize the best continuation at every intersection, we can define the term *decision point*. A decision point is a point on a route where the user needs to change the direction of travel (reorientation). An intersection with only two outgoing edges is not a decision point for the reason that the user can only continue onward (we assume that the user does not decide to turn back). An intersection where the user follows the best continuation is also not a decision point because we assume that it is improbable that the user chooses a path other than the best continuation. In all other cases, a decision instruction with a maneuver must be used so that the user does not deviate from the route.

We also quantify the need for the decision instruction by defining a prior probability $P_p(e_j|e_i)$ that gives the probability that the user approaching the intersection along the edge e_i will choose the edge e_j without providing any information. The value of the prior probability is defined in Formula (1).

$$P_p(e_j|e_i) = \begin{cases} 1 & e_j \text{ is the best continuation} \\ \frac{1}{|E_{straight}|} & \text{in case of multiple straight continuations and no best continuation} \\ 0 & \text{in case there is no straight continuation} \end{cases} \quad (1)$$

3.2.6 Route

Using the previous definition of a path 3.11 and an itinerary 3.17, the route is defined as follows.

Definition 3.18. A *route* from vertex v_s to vertex v_e is a tuple $P = (W, \mathbf{I})$, where W is a path and $\mathbf{I}$ is an itinerary such that $W = (v_1, e_1, \ldots, v_n)$, $v_1 = v_s$ and $v_n = v_e$.

3.3 Objective

The main objective of this work is to generate an itinerary with enriched instructions. Since the itinerary depends on the underlying path, we also try to optimize the path so that the route is easy to describe.

3.3.1 Optimal itinerary

We first describe the goal of finding the optimal itinerary for a fixed path. In this subproblem, we need to determine which instructions will be used and which will be omitted. This problem may resemble a scheduling problem with the difference that in scheduling problems, we have to schedule out all the tasks and determine their starting times. In this case, we have a fixed starting time for each instruction and decide which instructions to use. This subproblem is utilized during the evaluation (Chapter 6), where we evaluate the performance on a defined set of paths.

The optimal value is the cost of the path **W**, denoted as $c(\mathbf{W})$ and it represents the probability that the user will successfully complete the path without any deviation from the path.

We describe our goal of finding an optimal route for a fixed path as an optimization problem, shown in Figure 3.2.

$$\max \quad c(\mathbf{W}) = A + \varepsilon \times \frac{1}{n} \sum_i s_i$$

subject to

$$A = \prod_{(e_i, e_{i+1})} P_p(e_{i+1}|e_i)^{1 - s_{(e_i, e_{i+1})}} \qquad \forall e_i, e_{i+1} \in \mathbf{W}$$

$$\forall I_{(x,y)}, I_{(z,w)}, x, y, z, w \in \mathbf{W}, (x,y) \neq (z,w) \wedge D^{(x,y)}_{start} \leq D^{(z,w)}_{start}:$$

$$D^{(x,y)}_{start} + D^{(x,y)}_{processing} < D^{(z,w)}_{start} + M * (2 - s_{(x,y)} - s_{(z,w)})$$

Parameters:

P_p ... prior probability that the user does not deviate from the path at the vertex $v_i = e_i \cap e_{i+1}, v_i \in \mathbf{W}$, when **not** given the associated navigation instruction $I_{(e_i, e_{i+1})}$ (i.e. will choose e_{i+1} at e_i)

M ... big M constant $\left(\text{for example, } \max_i(D^i_{start}) + \max_j(D^j_{processing})\right)$

W ... path, as defined in 3.11

ε ... weight representing the weight of the instruction count

$-1 < \varepsilon < 0$: minimize the instruction count

$0 < \varepsilon < 1$: maximize the instruction count

Variables:

s_i ... binary variable that indicates if the instruction I_i is used (1) or not (0), where $i = (x, y)$ and $x, y \in \mathbf{W}$

Figure 3.2. Optimal itinerary

3.3.2 Optimal route

Now that we can calculate the cost of any optimal itinerary for a given path, we can define the optimal route (denoted P^*) for all possible paths between the start and end points. This optimal route should consist of a path that can be easily described with a high probability that the user will successfully complete the route.

$$P^* = \max_{\mathbf{W}} c(\mathbf{W})$$

On the grounds that the aim of this work is to generate an (optimal) itinerary of enriched instructions, we have focused first on maintaining the optimality of the itinerary and, in the second place, on finding the path that is close to the optimal route.

Chapter 4
Solution Approach

In this chapter, we explain the details of the problem specification. First, we refine the definition of the landmarks by describing our method of obtaining them. Since landmarks are not the only type of information, we also describe the features of our routing graph. We then describe our instructions, including their text representation based on available information and how the types of instruction differ. In the last step, we propose a set of operations to integrate each instruction into an itinerary of instructions.

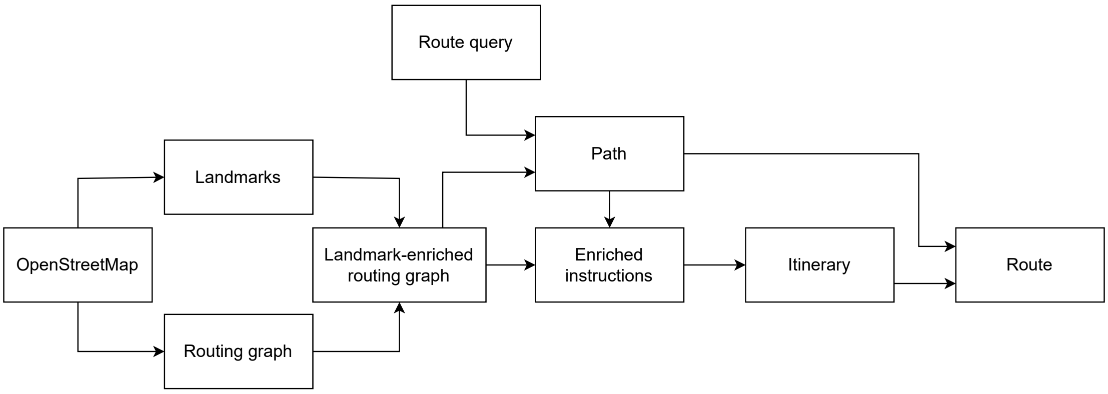


Figure 4.1. Steps of our solution approach

Our solution approach is visualized in Figure 4.1 which shows all the steps of our method. The node *Route query* that represents a routing request between two points can be omitted in case we already have some fixed path to which we need to find the itinerary of instructions.

4.1 Landmark extraction

In this section, we describe both landmarks (definition 3.1.5), features of vertices and edges in landmark-enriched routing graph (definition 3.1.4), and how these types of information are extracted from maps. We describe our search perimeter, the location of landmarks relative to the user's position, and the factors that influence the landmark salience. The salience function is used to select the most prominent landmark near the user that will be used in the instructions.

4.1.1 Landmark neighborhood

The search perimeter (from definition 3.1.3) is a parameter of our approach that determines the neighborhood (boundary) where the object must be located to be considered a nearby landmark. The size was determined to be 50 meters based on previous works [2, 26], which have also opted for this exact value. The reason for this value is that landmarks beyond this distance are more likely to be obstructed or not relevant to the route. The benefit of keeping the perimeter size low is the computational speed, as the number of landmarks grows quadratically with increasing perimeter size. Lower values may, on the other hand, omit some useful landmarks in less dense areas.

For each direction and type of instruction, a different landmark may be suitable. We define five points along the edge geometry: the midpoint, the source point, the target point, the source reference point, and the target reference point. All these points are visualized and shown with a nearby landmark in Figure 4.2. The reference points and midpoint are used to calculate the distance between the user and the landmark, and the source and target points are used to determine the relative location of landmarks (Section 4.1.2).

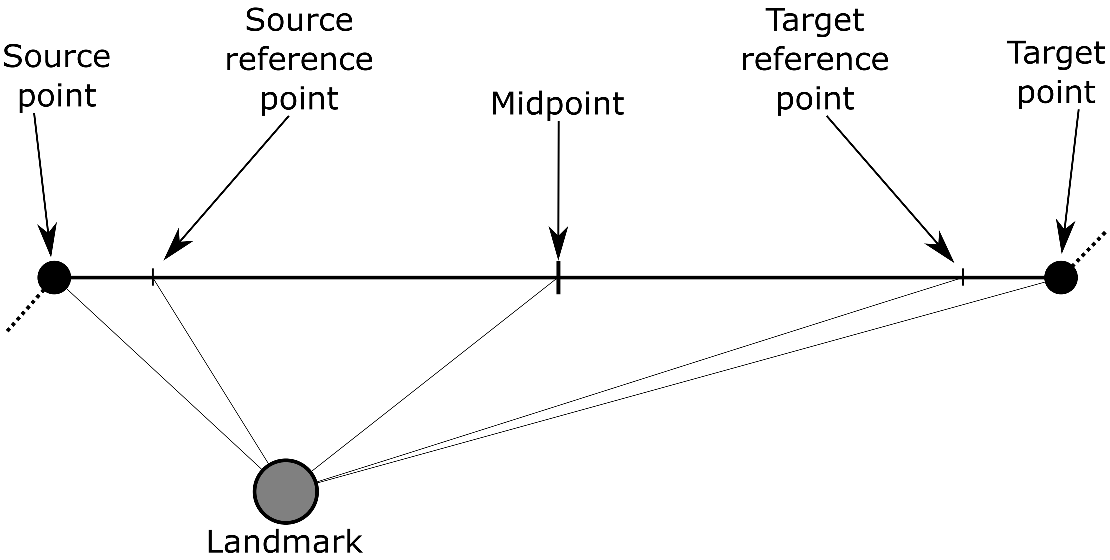


Figure 4.2. Points located on the geometry of the edge

Reference points are defined as points located D units away from the source or target point, respectively, where D is the size of the search perimeter. If the edge is shorter than $4 \times D$, then this point is in the first or third quarter of the total length from the source point. These reference points are used so that the visibility of all nearby landmarks is checked in advance (the user approaches the source or target intersection, in case of the approach instruction) or from the intersection (the user heads away from the source or target intersection, in case of the decision instruction). This idea of offsetting reference points was taken from articles [2, 8], which use the terms *reference point* or *advance visibility*.

If the distance from a landmark to the midpoint is less than the distance to both reference points, the landmark can be used in the confirmation instruction, as the confirmation instruction refers to a landmark along the route. If the landmark is closest to the source or target reference point, the landmark can be used in the decision or approach instructions. If the landmark is closer to the source reference point, it is considered for the approach instruction at the source node (the user is approaching the source node from the target node) and the decision instruction at the source node (the user being in the source node and deciding on the next continuation). If the length of the edge is shorter than D, the landmark can be attributed both to the source and the target point regardless of the distance. The reason is that both ends are very close to the landmark, and the landmark is useful in both cases at the cost of the landmark being duplicated at both ends of the edge.

4.1.2 Relative location of landmarks

An important factor that influences the overall salience of the landmark is the position relative to the decision point (intersection). This work distinguishes three possible locations: the landmark lies **before** the user's position (in front of the user), **at** the user's position, or **after** the user's position (behind the user). All these cases with respective angles are illustrated in Figure 4.3. This method is similar to the one published in [13]. The difference is that in this work the angles are not uniform, and the preposition *at* is rarely used in favor of the other two. The reason for this split of angles is to maintain consistency with the angles of the maneuvers (described later in Section 4.5.1).

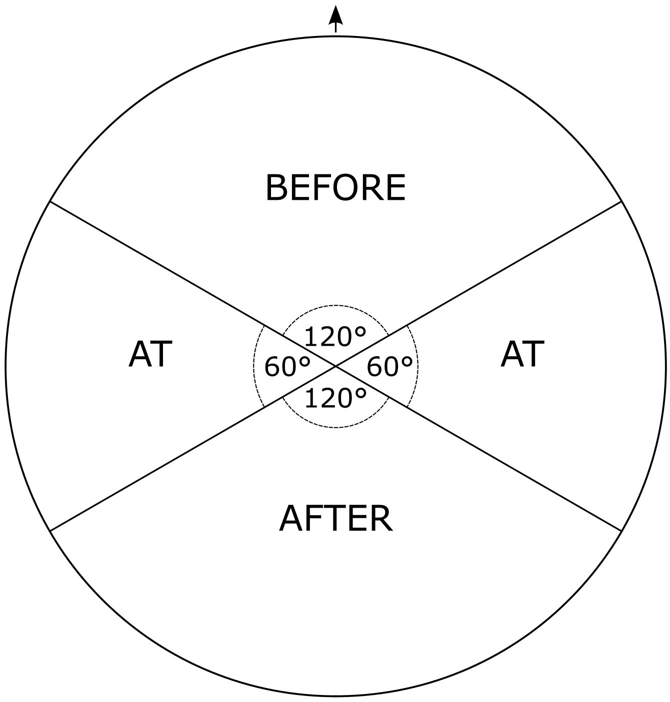


Figure 4.3. Prepositions for landmarks based on their relative location. The top arrow shows the relative north (relative bearing).

4.1.3 Determining landmark visibility

Some landmarks may not be visible from the current path, so the visibility of all landmarks must be considered. This is checked during the landmark search (described later in Section 4.4). When a nearby object can be both a landmark and an obstacle (e.g., a building), this nearby object is deemed to be a visible landmark if there is no other obstacle in the line of sight.

A problem might occur when a landmark is enclosed by another landmark or an obstacle. Buildings can contain points that mark the location of shops or other utilities located inside the building, or the purpose or function of the building is expressed by this inner point. In both cases, we assume that these points are not obstructed by the enclosed building and are visible from the outside, and therefore we do not include this building in the set of obstacles for the inner points. Compared to the method shown in work [2], our approach is more straightforward but may falsely accept a point landmark, which is located on the other side of a building.

Even after this simplification, there still exist some cases where this approach may falsely reject a visible landmark. This case is shown in Figure 4.4. In this case, moving the landmark to the closest edge would not help as it would be moved to that side of the building, which is not visible. An improvement of our approach would be to use the whole geometry of the building instead of the landmark's own geometry. This may, on the other hand, create some other problems, such as multipurpose buildings with more than one inner landmark (shopping centers) or a large building where the landmark occupies only a small portion of the building's space. This problem can be summarized as the determination of the relationship between the building and its inner landmarks.

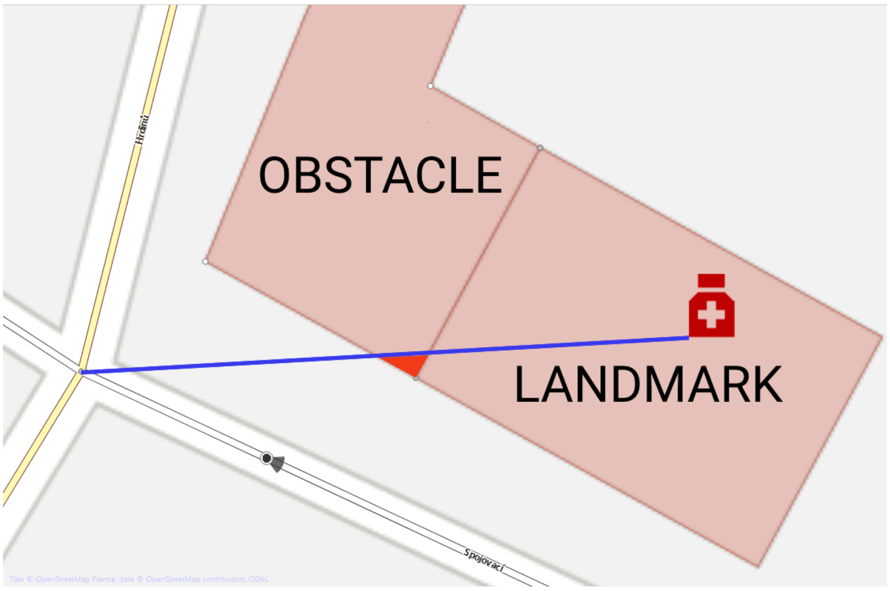


Figure 4.4. Falsely rejected landmark due to the corner of a nearby building acting as an obstacle (Source: OpenStreetMap)

In all cases, we compute the shortest line connecting the reference point to the landmark (which can be thought of as a line of sight) and check if it is blocked by any obstacle. If the landmark is blocked, we can either remove this landmark or apply only a high penalty and keep it as a landmark. If we refer to a landmark that is not visible, we might confuse the user or the user may choose the wrong path. We have thus chosen the former approach with a slight risk of removing a visible landmark (false negative) that may have been useful to the user during the wayfinding.

4.1.4 OpenStreetMap data

The spatial data used for this thesis are from the OpenStreetMap[1] project, which is built by a community of mappers that contribute and maintain data about roads, paths, shops, and much more all over the world.[2] Anyone can contribute to the project by extending, altering information, or removing outdated data. Data quality may differ in various parts of the world, as reported in article [2] by A. Rousell and A. Zipf.

This dataset contains a wide range of objects that might be suitable for enriching instructions and has already been used in the past in numerous works (articles [2–3, 13–15]).

[1] https://www.openstreetmap.org/
[2] https://www.openstreetmap.org/about

4.1.5 Landmark categories in OpenStreetMap

In OpenStreetMap, landmarks can be any object, ranging from single nodes (fountain, memorial, tree), ways (waterway, road, railroad), or areas (building, park, church). Not all landmarks are equally useful for navigation instructions. In most cases, a large church should provide a better waypoint for orientation and wayfinding purposes than an indistinct building in a city area.

Based on the articles [2, 13], we have split all landmarks into categories with weights, chosen similarly to these articles. Although the exact weights of the categories in these two articles vary, the relative order of these categories is maintained. Each landmark category is described using the same key and value pair as defined in the OpenStreetMap Wiki[3] and the weights chosen for the landmark categories are shown in Table 4.1. The * symbol in the column *Values* denotes that any value is matched (and only the key matters).

4.1.6 Obstacles in OpenStreetMap

Table 4.2 shows all categories of objects that we consider to be obstacles in OpenStreetMap. The other problem related to categories is that we cannot detect a landmark that should not be visible. Reasons for this include that the obstacle is not in the dataset or that the category of the obstacle is not viewed as an obstacle class. We have omitted the fences for the reason that the user may still see a possible landmark through the fence or the fence is part of the landmark (school area). This problem is caused by our approach, where we do not consider obstacles to be semi-transparent. Examples of semi-transparent objects are village green, hedges, fences, or trees. The landmark on the opposite side of the road is a similar case and is also unfavorable, but our approach does not recognize and penalize this special case.

Key	Values
barrier	city_wall, hedge, retaining_wall, wall
wall	*
highway	motorway, trunk, primary
	motorway_link, trunk_link, primary_link
building	*

Table 4.2. Key and values of obstacles

4.1.7 Salience

Each node and edge have multiple valid and visible landmarks. Similar to the previous articles [2, 7–8, 13], we define a salience function, with which we are able to compare and score each landmark based on its properties. The salience function is influenced by the distance from the node or edge, the category of the landmark (Section 4.1.5), location of the landmark relative to the node or edge (Section 4.1.2), area of the landmark, if the landmark has a name, the length or area of the landmark, the uniqueness (number of landmarks with the same category), and if it is a relation (composed of more than one object, a specific class of objects in OpenStreetMap dataset). The salience formula is shown in Formula (1).

[3] https://wiki.openstreetmap.org/wiki/Map_features

Weight	Key	Values
1.00	amenity	place_of_worship
	building	cathedral, chapel, church, mosque, synagogue, temple
0.91	amenity	fountain, fuel
0.90	amenity	police, fire_station
0.85	amenity	cinema, theatre
0.84	tourism	hotel
	amenity	restaurant, pub, cafe, bar, post_office, fast_food
0.83	railway	subway_entrance, subway_station, tram_stop, station, level_crossing, tram_level_crossing, tram_crossing, crossing
	bridge	*
	man_made	bridge
0.81	shop	supermarket
0.80	historic	memorial, monument, statue, wayside_cross, way-side_shrine, castle, church
0.75	amenity	school, university, pharmacy, hospital, clinic, library, college
	building	school, university
0.70	leisure	park, playground, pitch, stadium, sports_centre
0.65	shop	bakery, chemist, jewelry, kiosk
0.45	amenity	embassy, bank, courthouse, town_hall
	diplomatic	*
	office	*
	historic	*
	tourism	museum, artwork
0.42	shop	*
0.30	public_transport	*
	amenity	bicycle_rental, bus_station
	substation	minor_distribution, distribution, transmission, transi-tion, generation, converter, measurement
	power	tower, pole, generator, substation, transformer, switch, plant
	man_made	antenna, tower, pier, storage_tank, silo, works, wa-ter_tower, chimney
0.25	vending	*
	recycling_type	*
0.20	amenity	*
	club	*
	highway	traffic_signals, bus_stop
	crossing	traffic_signals
	landuse	*
	garden:type	botanical, castle, community, monastery
	water	*
	waterway	*
0.1	building	*
	barrier	*
	railway	rail, tram, subway

Table 4.1. Categories of landmarks and their weights.

Other factors specific to the OpenStreetMap dataset (Section 4.1.4) that we have not used are the age of the OpenStreetMap object (newly added objects are more likely to be up to date), or the number of tags the object has (more significant objects have opening hours, contact info, and other useful properties). More features of objects that may be useful can be seen in article [17], where the authors use data mining methods over their proposed features.

$$salience = position + \frac{D - distance}{D} + weight + name + \frac{1}{name\ length} + relation + \frac{1}{area} + \frac{1}{same\ landmarks} \quad (1)$$

- $position$: weights for relative positions
- $distance$: distance from the path
- D: search perimeter size
- $weight$: weight of the category of the landmark
- $name$: if landmark has a name (1) or not (0)
- $name\ length$: length of the name in characters (0 for a missing name)
- $relation$: if landmark is a relation (0) or not (1)
- $area$: area of the landmark, or length (in case of a line)
- $same\ landmarks$: number of landmarks being in the same category, where $\frac{1}{same\ landmarks}$ is uniqueness of the landmark category

The location of the landmark relative to the node or edge influences its suitability (Section 4.1.2). For approach instructions, we prefer landmarks that occur before the intersection (the user passes this landmark). For decision instructions, the landmarks that occur after the intersection (in front of the user) are preferred. For confirmation instructions, the relative location is not as important as the other factors (distance, category) as long as the landmark is visible at some point along the edge. The weights are inspired by the weights from article [2] with the difference that in this article the authors have decided to apply the relative position as a factor (multiplier) of the entire salience term (the term is multiplied by this weight). For an approach instruction, this means that a landmark before the intersection is three times more salient than a landmark that appears after the intersection. Although the reason is described in the literature (landmarks before intersection should be strongly preferred), we have applied the weight of relative location linearly (added to the whole term) to weaken the overall effect of this criterion. The resulting weights are shown in Table 4.3. In all definitions, the salience function enables comparing landmarks and selecting the most salient ones.

	Before	At	After
Approach	1	2	3
Decision	3	2	1
Confirmation	1	1	1

Table 4.3. Weights of landmark's relative position for different types of instructions

4.1.8 Graph features

The second type of information is the feature of the vertices and edges (definition 3.1.4) in our routing graph. A problem arises when the instruction contains both landmarks and features, each of which has a different quality. To keep the instructions as brief as possible, we have to choose only one of them. Thus, we extend the salience function of landmarks to features. The exact salience values are shown in Table 4.4.

Information	Salience
Junctions	
T junction	$\frac{1}{2}$
Y junction	$\frac{1}{3}$
Four-way junction	$\frac{1}{6}$
Road type	
Stairs	1
Bridge	1
Pedestrian crossing	0.99
Sidewalk	0.5
Uphill or downhill	0.25
Class of road (with or without name)	0.1

Table 4.4. Salience values for features

We detect the types of junctions by the number of outgoing edges and their angle relative to the incoming edge. The Y junction is a junction with three edges (two outgoing and one incoming), where one outgoing edge leads slightly left, and the other leads slightly right. The T junction (defined similarly to the Y junction) has three edges, with one being the incoming edge and the other two leading to the left and right. The four-way junction is a junction where the user can continue straight ahead, go left, or go right.

Although we have defined our space as two-dimensional, height information can be useful for describing the properties of edges (edge leading up or downhill). The OpenStreetMap data do not contain information on the height of points, so we additionally use the SRTM dataset[4] with 3 arc second resolution from NASA as our height model. This dataset is processed with the SRTM Plugin for OpenStreetMap's Osmosis[5] (a plugin for the Osmosis tool[6]). This plugin adds height data for each node (vertex) in the OpenStreetMap data and outputs a modified file with all other features unmodified. Even though an SRTM dataset with 1 arc-second resolution is available, we have chosen this less accurate dataset for the reason that the SRTM Plugin supports only SRTM datasets with this exact resolution.

The features of the vertices and edges are extracted from OpenStreetMap using the key and values of the objects. The types of roads are derived from tags of objects. Stairs are edges with the tag `highway=steps` and optionally combined with the tag `incline=up` or `incline=up`. If the `incline` key is missing, we use the height difference from the SRTM dataset to determine the incline. Bridges are edges with the key `bridge` and can have any value (mostly with the value `yes`[7]). Bridges can also be

[4] https://lpdaac.usgs.gov/products/srtmgl3v003/
[5] https://github.com/locked-fg/osmosis-srtm-plugin/
[6] https://wiki.openstreetmap.org/wiki/Osmosis
[7] https://taginfo.openstreetmap.org/keys/bridge#values

treated as a landmark when the route leads under them (mentioned later in Section 4.3). Pedestrian crossings and sidewalks (`pedestrian=crossing/sidewalk`) are edges with the tag `highway=pedestrian` and are used when the type of road changes. Other changes to the class of a road are also used but are usually ignored in favor of a landmark.

For each edge, we calculate the angle of inclination using the height difference between the end and start point and the length of the edge ($arctan(\alpha) = \frac{\Delta height}{length}$). If the absolute value of the angle α is greater than some threshold ($|\alpha| > \Theta$), we consider this edge leading uphill or downhill. Based on expert knowledge[8], we have chosen the threshold value Θ as 6.3° or 10%.

4.2 Creating a routable graph

The data obtained from OpenStreetMap are created by people and were not created to be routable out of the box, so some paths may cross each other without having a valid intersection point. In work [26], the function `pgr_nodeNetwork` from the pgRouting[9] library was utilized for this task. The significant disadvantage of this approach was that at most one intersection was created between each pair of edges. As a result, some paths were not properly connected and the resulting network was not sufficient for routing purposes.

For these reasons, we have chosen the osm2po[10] tool, which reads OSM files directly and produces a routable graph that is properly noded. The behavior can be modified using a configuration file that specifies the types of roads with respect to the access conditions (roads that are not public or highways are omitted). The default configuration file is suitable for car routing, we have thus slightly modified the configuration to support bicycle routing by adding cycleways, pedestrian roads, sidewalks, crosswalks, and stairs with corresponding access conditions. This tool can also output an SQL file representing the processed (routable) graph that can then be imported into a PostgreSQL database (with a PostGIS extension) with a table structure similar to that generated using the pgRouting library. After these steps, the database table containing the data is ready to be processed, and all extracted paths can be enriched with information about landmarks.

Although the osm2po tool is freeware with an open license (shown after running this tool), it is neither a free software[11] nor an open-source project. Therefore, we cannot inspect the internal structure or reuse any of its source code for this project.

4.3 Building landmark-enriched routing graph

We have reused a large portion of the source code from our previous work [26] for the routing graph enrichment, with significant modifications and additions made to the functions used for outputting edges enriched with landmarks.

[8] https://wiki.openstreetmap.org/wiki/Key:mtb:scale,
https://theclimbingcyclist.com/gradients-and-cycling-an-introduction/,
https://www.cyclist.co.uk/in-depth/682/how-steep-is-too-steep-when-cycling-uphill
[9] https://pgrouting.org
[10] https://osm2po.de
[11] https://www.gnu.org/philosophy/free-sw.en.html

The data from the osm2po and osm2pgsql tools are imported into a PostgreSQL database, as described in the previous Section 4.2, and processed using a Java application. It uses a wide range of libraries for this task, mainly the JDBC driver for communication with the PostgreSQL database, the Java Topology Suite (JTS)[12] for handling geometric data, the Geotools library for working with spatial data, Google GSON[13] for serializing edges into a GeoJSON file, and other libraries (described later in Section 5).

First, all edges in the routable graph are loaded, including information about their ID (assigned by the osm2po tool), ID from OpenStreetMap, geometry, identificators of source and target nodes, class of the edge (with details about footway type or being a bridge and its incline), layer, information about being a one-way edge, and the relative elevation between the source and target node.

Each edge of the graph is loaded and processed separately (in parallel using the Java Stream API), and the result is then outputted into a GeoJSON file. For each edge, we need to gather information on the landmarks of the approach, decision, and confirmation instructions in both directions. The suitability of the landmark for different situations based on the distance from the edge or the vertex is described in Section 4.1.1.

In addition to the distance between the landmark and the edge, we also calculate the angle relative to the edge to determine the spatial relationship. The landmark can be located before, at, or after the point (Section 4.1.2). Bridges, crossings (pedestrian, railway), waterways, or areas are landmarks that the user can come across, and we need to detect the height relationship between the edge and the landmark. We have used the **crosses** predicate (function) from the Dimensionally Extended 9-Intersection Model (DE-9IM)[14]. This predicate is true if at least one point from the edge intersects the landmark and not just one of the end points. Due to numerical stability, we have shrunk each edge by 1 meter to avoid any possible problems. Based on this predicate, we are able to generate instructions such as “Continue under the bridge” or “Turn left over the railway crossing”.

Among the collected landmarks can be those that are obstructed by some obstacle (Section 4.1.3), which are removed and not considered. From all these collected landmarks, a single landmark has to be selected for each direction and type of instruction. The criterion (score) for the comparison of landmarks is their salience, as defined in Section 4.1.7. The processed edge includes five landmarks: landmarks for the approach instruction in both directions, the decision instruction in both directions, and the confirmation instruction in the direction from the source to the target node.

Each edge is then outputted into a GeoJSON file containing each edge’s geometry, their properties (ID, class, source and target nodes, name, landmarks, etc.), and the referenced landmarks that contain both their geometry and properties (ID, category, area). The advantage of this format is human readability (compared to binary file formats), having a standardized scheme, being a commonly used format for spatial data (compatible with most geospatial tools), and the ability to work with JSON via JavaScript in the majority of modern web browsers. The most significant disadvantage is the file size, because the raw file needs to remain readable, and the keys in objects often need to be repeated. This disadvantage can be overcome by compressing the output file, either by compressing the file or by using compression in the HTTP protocol (if the whole file needs to be transmitted).

[12] `https://locationtech.github.io/jts/`
[13] `https://github.com/google/gson`
[14] `https://en.wikipedia.org/wiki/DE-9IM`

This graph enrichment process ends after the GeoJSON file is outputted. This landmark-enriched routing graph is then used during the instruction generation (Section 4.5) and graph routing (Section 4.4).

4.4 Path planning

There are two possible ways to obtain a path. We can either use a precomputed path for which we generate the itinerary or we can find a path that is easy to describe. This section will describe the planning algorithm for finding such an optimal path, which should be easier to describe than the shortest path.

Building the enriched graph is a step that we take only once for the entire map area, but finding the optimal itinerary and optimal path must be done for every route search (query).

The first step before constructing the graph is to ensure that the edges in this graph are part of one graph component, so that there would be a path between any two vertices. Therefore, we first search for the largest graph component using the breadth-first search (BFS) algorithm and then use only this largest component. All edges and vertices that are not part of this component are discarded.

The next step is to construct a graph (a component of the original graph) with enriched edges and connect the adjacent edges with maneuvers. Each pair of adjacent edges is described using a maneuver (described in Section 4.5.1) that is sufficient in most cases to fully describe the overall action (reorientation). In some cases, the maneuver can be omitted as the user just continues straight ahead and does not change the direction of travel. This best continuation is described in detail in Section 4.6 that is related to the prior probability P_p that the user will select any outgoing edge given that we know the incoming edge (direction of the approach), as described in Section 3.2.5. This probability is vital for selecting a route where the user has the lowest chance of getting lost.

A path is found in the graph using Dijkstra's algorithm, where the graph is expanded over edges instead of nodes, which is similar to an edge-based graph. The difference from the edge-based graph method is that we did not apply a transformation from our edge-based graph to a node-based graph and we expand each edge dynamically during the search. The reason for this way of search expansion is due to the definition of the prior probability $P_p(e_j|e_i)$ inside the intersection, which is defined on two consecutive edges. In this case, we need to apply costs to the next edge depending on the incoming edge.

The cost of each tuple of consecutive edges is influenced by their length, the prior probability (continuing straight is cheaper), the class of the road (the user may prefer to avoid secondary and tertiary roads or pedestrian ways), and if the edge leads in the opposite direction of a one-way street (the edge is traversable at a higher cost). Since each user may have slightly different preferences, these parameters can be modified for each routing request (described later in Section 5.3). Another parameter is the user's speed, which determines the density of instructions in every route leg (edges between decision nodes). When the user travels at a lower speed, there is enough time to announce more instructions, and at a higher speed, the itinerary consists mainly of the decision instructions. These parameters are also contradictory to each other, as the most simple path is usually not the best for cyclists, because the simplest path usually follows the car infrastructure.

Alternatives to our routing approach include a turn table method, or an extended cycleway graph. All these alternative methods are described in more detail in [29]. Since computationally efficient routing was not the main goal of this thesis, we have not utilized some advanced methods, such as contraction hierarchies, routing profiles, or bidirectional search.

4.4.1 Node spatial index

When the user wants to plan a route between two geographical coordinates, we need to provide a way to snap the start and end coordinates to the nearest node in the graph. A naive approach would iterate over all nodes, calculate the distance difference, and select the closest node. Fortunately, the JTS library contains an STR tree (Sort-Tile-Recursive tree) to query two-dimensional spatial data. Using this method, we are able to find the nearest neighbor node for any coordinate more efficiently.

4.5 Generating instructions

We have already described the types of information that each instruction can use. In this section, we describe the text templates of the instructions using formal grammar in BNF (Backus–Naur form), shown in Figure 4.5. We have simplified the definition of the nonterminal variables `<landmark-type>` and `<name>` for clarity. The `<landmark-type>` corresponds to the name of the category of the landmark (Section 4.1.5) and the `<name>` is the name of the landmark or the road.

The distance nonterminal is shaped to allow only numbers that start with a nonzero digit and end with a zero (rounded to tens of meters). By limiting the number of digits in the distance rule, it can be easily extended to kilometers. We also do not deal with the capitalization of the first letter in instructions, making them proper sentences. The limitation of formal grammar is that it does not capture the logic that we use to select the proper substitution for each nonterminal (choosing the most helpful information). Consequently, the instructions generated by this grammar are syntactically correct but not necessarily logically correct ("continue under the school" is an undesirable instruction).

The BNF grammar can be examined using the site `https://bnfplayground.pauliankline.com/` that can verify already generated instructions or generate new random instructions.

The same idea can also be expressed using a flow chart which is shown in Figure 4.6. The advantage of the flowchart is its comprehensibility compared to formal grammar. Formal grammar can, on the other hand, be more convenient for algorithmic processing. An application of this grammar would be to check if each instruction in the itinerary is a valid instruction based on the grammar templates.

4.5.1 Maneuver

When the user is at the intersection, the user needs to choose the correct continuation based on the planned route. Traditionally, this is solved by announcing a maneuver, which is just ordinary information about the change of direction relative to the user's current heading. The user is instructed to either continue straight ahead, go slightly left or slightly right (or otherwise called bear left or bear right), go left or right, go sharp left or sharp right, and go back (as defined in 3.13). Although the resolution of these maneuvers should be sufficient, it is likely that multiple continuations of the current path lead in a similar direction so that the user might get confused, may choose

```
<instruction> ::= <approach_instr> | <decision_instr> |
        <confirmation_instr>

<approach_instr> ::= <approach_info> " " <maneuver> |
        "the target is " <approach_info>
<approach_info> ::= "after " <distance> " meters" |
        "at the " <junction> | <approach_landmark>
<approach_landmark> ::= "at " <landmark> | "before " <landmark> |
        "after " <landmark>
<junction> ::= "Y junction" | "T junction" | "four-way junction"
<decision_instr> ::= <decision_instr_now> |
        <decision_instr_now> " and then " <decision_instr_next> |
        <decision_instr_now> " and then continue for "
        <distance> " meters"
<decision_instr_now> ::= <maneuver> | <maneuver> " " <decision_info> |
        "go " <decision_info>
<decision_info> ::= <decision_landmark> | <road_info> | <compass_heading>
<decision_landmark> ::= "towards " <landmark> | "by " <landmark> |
        "away from " <landmark>
<decision_instr_next> ::= <maneuver> | "continue " <decision_landmark> |
        "continue " <road_info>
<confirmation_instr> ::= <confirmation_landmark> |
        "continue " <road_info>
<confirmation_landmark> ::= "continue over " <landmark> |
        "continue under " <landmark> | <landmark> " is on your left" |
        <landmark> " is on your right" | <landmark> " is in front of you"

<landmark> ::= "a " <landmark_type> | "the " <landmark_type> " " <name>
<landmark_type> ::= "school" | "church" | "shop" | "bus stop"
<name> ::= "NAME"
<road_info> ::= "onto the " <road_class> | "onto " <name> " Street" |
        "over the bridge" | "over the bridge " <name> |
        "up the stairs" | "down the stairs" |
        "over the crosswalk" | "onto the sidewalk" |
        "uphill" | "downhill"
<road_class> ::= "secondary road" | "tertiary road" | "cycleway" |
        "unclassified road" | "residential road" | "service road" |
        "pedestrian way"
<maneuver> ::= "continue straight" | "turn left" | "turn right" |
        "turn back" | "turn slight left" | "turn slight right" |
        "turn sharp left" | "turn sharp right"
<compass_heading> ::= "north" | "south" | "east" | "west" |
        "northeast" | "northwest" | "southeast" | "southwest"
<distance> ::= <digit_nonzero> "0" | <digit_nonzero> <number> "0"
<digit_nonzero> ::= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
<number> ::= "0" | <digit_nonzero> | <number> <number>
```

Figure 4.5. BNF grammar for instruction templates (type of landmarks and name nonterminals are simplified for clarity)

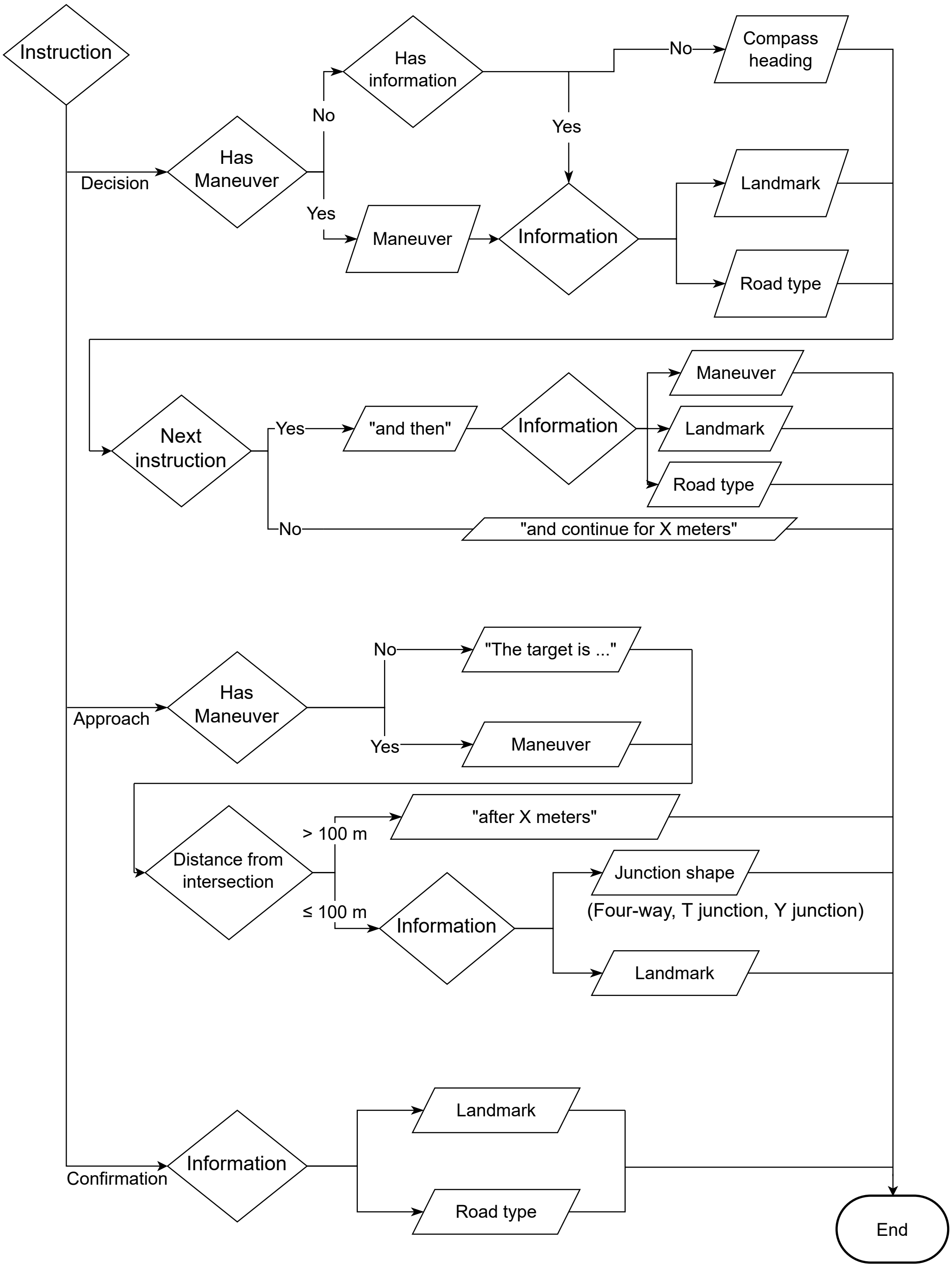


Figure 4.6. Flowchart for instruction generation

the wrong path, and possibly get lost. All maneuvers for edges that are not the best continuation are shown in Figure 4.7. The angles are chosen in a way similar to that in [13] with the difference of omitting the straight-ahead maneuver, which we assign only to the best continuation.

4.5.2 Approach instruction

The approach instruction is used when the user needs to identify the correct intersection while traveling along the way. In conventional navigation systems, this is solved by

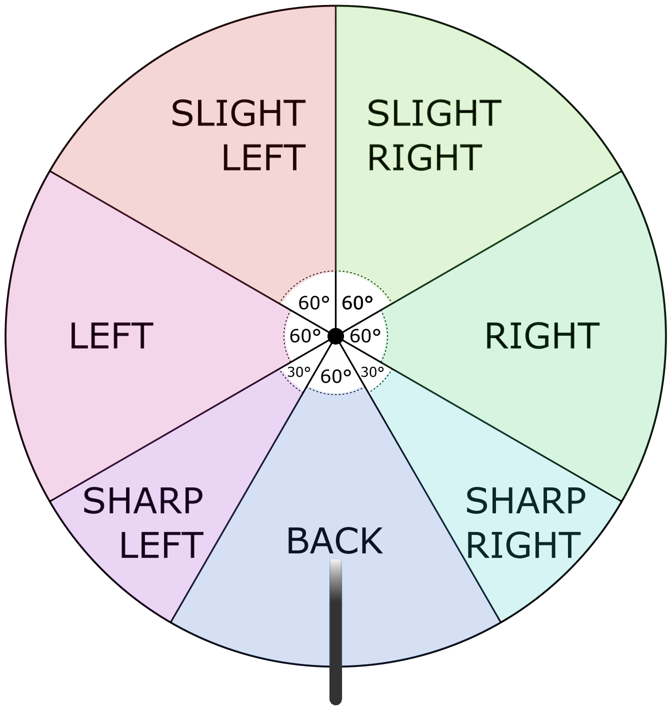


Figure 4.7. Types of maneuvers for edges that are not the best continuation

informing the user about the distance from the selected intersection ("After 100 meters turn left"). This information may fail in cases where there are at least two intersections (decision points) close to each other, and the user cannot safely determine the intersection referred to by the navigation instruction. This can be solved by informing the user about a landmark close to the intersection and its position relative to the current approach direction ("At the restaurant Budvarka turn slight right").

When there is no nearby landmark, we can substitute it with another piece of information. The junction itself is considered a landmark (in article [7] regarded as a structural landmark). Each junction can be described by its property (with traffic lights, pedestrian crossing), its shape (four-way intersection, T junction, Y junction), or by its functional property ("At the end of the street"). Additionally, we can also describe the change in the environment around the intersection ("After you leave the park").

We can omit the approach instruction when the adjacent decision instructions (the previous and the next one) are close to each other and there is no time to announce the approach instruction in advance. The only approach instruction that cannot be omitted is the last instruction that describes the target end point. The last approach instruction also does not contain any maneuver, as the user just ends the route at the described intersection. In the last instruction, we need to specify the location of the target waypoint, either by the shape of the junction, using a reference to some nearby landmark, or by announcing the distance from the target waypoint.

When an approach instruction is announced far enough from the intersection, we provide the information on the metric distance to the intersection to the user. An alternative would be to use time information based on the user's speed ("After 5 minutes turn left"). In the case of a pedestrian, we may even consider using the number of steps as a measure of distance ("After 200 steps turn sharp right").

4.5.3 Decision instruction

In some cases, a maneuver is not enough to communicate the desired change of direction to the user. The decision instruction incorporating information about a nearby landmark should help the user make the right decision in ambiguous situations by leading the user towards, by, or away from nearby landmarks. A special case occurs when the user starts their journey inside a junction and needs to head in the right direction. In this case, a maneuver cannot be used, as it relies on knowing the current heading of the user, and other information needs to be provided to navigate the user along the computed route. We can provide either some landmark towards which the user should head, the type (class) of the road that the user should use, or the compass heading in case all other types of information are missing.

The decision instruction can also refer to some properties of the outgoing edge. Such properties can be the class or type of edge (road, cycleway, crossroad, stairs, bridge), height difference (uphill or downhill), name of the street, etc. If the instruction contains information on both the landmark and the road features, we keep only the one that provides more utility to the user to keep the instruction simple.

After that, we usually need to inform the user what to expect after this intersection. We can provide information about the length of the oncoming part of the route ("Turn right and then continue for 500 meters"), about the next maneuver in case the second maneuver needs to be executed right after the first one ("Turn left and then turn right"), or about the area through which this part of the route leads ("Continue through the park").

4.5.4 Confirmation instruction

When the user is going along some longer edge and the navigation does not provide any instructions for an extended period of time, the user might become unsure and doubtful about the correctness of the route. We need some information that the user can process without diverting the user's attention away from the route to the navigation device, where the route is being visualized. This can be solved by describing the environment of the current part of the route to the user by referring to some unique and distinguishing properties of the environment. This should reassure the user about the correctness of the path with minimal cognitive load and should reduce the need to check the path via the navigation device regularly.

Another source of doubt can be caused by a junction, where the user is expected to choose the best continuation (according to the algorithm described in 3.2.4), and no instruction is generated for this intersection. Although the user will most likely choose the correct continuation, some type of confirmation is needed so that the user can be reassured once again. Compared to the confirmation along the edge, we can refer to the intersection and its topology instead of nearby landmarks.

Related information can refer to some nearby landmark and its spatial relation to the edge ("School is on your left"), or it can describe a change in the class of the road without a change in direction ("Continue over the bridge"). The confirmation instruction is simpler compared to the decision and approach instruction, as it does not need to convey the information crucial to the successful completion of the route.

4.6 Implementation of the best continuation

As described in Section 3.2.4, the term *best continuation* is essential to predict which edge the user will choose without receiving any instruction. We have implemented a modified version of the algorithm defined in Wazeopedia[15] that describes how the Waze navigation system selects the best continuation for each intersection. This algorithm can be summarized as follows. Select all outgoing edges, where the angle between this selected outgoing edge and the incoming edge is smaller than 45 degrees, and the best continuation is an edge from this set of edges that has the strictly best matching relative to the incoming edge. For the best continuation, we can either give the instruction “Continue straight ahead” or omit the maneuver and skip the instruction entirely. There may be no such best continuation, in which case a maneuver has to be used for every outgoing edge. This approach is capable of generating maneuvers for a wide range of junction types, including difficult Y junctions or junctions more complex than just a typical 4-way intersection.

The situation with car infrastructure is usually easier because car roads are often divided into major and minor roads and generally have a lower branching structure. The cycling infrastructure can be more complex, so this method for best continuation may not always be suitable. This can be partially mitigated by using landmarks whenever possible or by also announcing instructions at decision points where the user continues straight ahead.

Our improved version of this selection process collects all possible candidates for the best continuation, and each candidate is assigned a score that is determined by the match of the type (class) of the road and the name of the road. These two attributes are decisive factors when most humans are trying to find the best continuation of the current road. Other factors that can contribute to this decision process would be marked cycle paths (which include not only cycleways), marked walking trails[16] or local traffic restrictions and street closures. If there exists a candidate that has the strictly highest score, then this candidate is deemed to be the best continuation. Otherwise, there is no best continuation, and for each outgoing edge, a decision instruction with a maneuver has to be announced to clear any ambiguity. This process for selecting the best continuation is shown in Formula (2).

$$\begin{aligned}
e_i &= (v_k, v_l) \ldots \text{edge, from which the user approaches the junction} \\
E_{out} &= \{e_j \mid e_j \in E \wedge IV(e_j) = v_l\} \\
E_{straight} &= \{e_j \mid e_j \in E_{out} \wedge |relative\ heading(e_i, e_j)| \leq 60^\circ\} \\
score(e_j) &= \begin{cases} 2 & \text{name and type is same as of } e_i \\ 1 & \text{name or type is same as of } e_i \\ 0 & \text{otherwise} \end{cases} \\
& e_j \in E_{straight} \text{ is the best continuation of the edge } e_i \iff \\
& \quad \forall e_k, e_k \neq e_j \colon score(e_j) > score(e_k)
\end{aligned} \tag{2}$$

One of the main differences is the choice of the angle that selects all straight continuations. In the original approach, an angle of 45.04° was used as the threshold angle. We have chosen a more obtuse angle (60°) to factor in irregularities in the bicycle and pedestrian road network and to match maneuver angles (Section 4.5.1), where the slight left and slight right maneuvers are used for edges that are not the best continuation in

[15] https://wazeopedia.waze.com/wiki/Global/How_Waze_determines_turn_/_keep_/_exit_maneuvers

[16] https://kct.cz/english

place of the straight-ahead maneuver. The other minor difference is that we take both the road name and the type as equal criteria, even though in the original approach, the name match is a stronger criterion.

4.7 Itinerary generation

In the previous section, we have proposed three types of instructions with different use cases and each of these instructions was created separately for each part of the path, one by one. For this reason, we propose a set of operations and functions to integrate (merge) these instructions into a single itinerary of consistent instructions.

The first step is to remove all meaningless instructions. We are not guaranteed that there is a landmark at every point of the route. Thus, the instruction may be reduced to an instruction that does not provide any new information to the user. An example of this would be the instruction “Continue straight ahead” because we assume that the user will continue if no instruction is provided. This rule does not apply to repeated approach instructions for the reason that the approach instruction usually announces a change of direction that is vital for the successful completion of the route.

The second step is to remove all duplicate instructions that are redundant and do not provide the user with any new meaningful information about the route. The question is in which way should we detect duplication of landmarks in the itinerary. The most straightforward method would be to compare the IDs of each landmark. The problem with this approach is that one real-world landmark can be marked using multiple objects. An example would be a public transport station, which may contain stop positions and platforms for each direction, and a relation that contains all previously mentioned objects. For this reason, we have chosen to compare each landmark using its textual representation, which consists of the type and the name (if present). The downside of this is that it removes landmarks of the same category in different places on the same route leg (a set of edges between decision points). Additionally, we also check the relative position of the landmark so that the landmark is announced again when its relative position changes. This enables us to create instructions such as “Go along the school building” and “After the school turn right”.

The next step, which may be seen as redundant, is a confirmation instruction at a complex intersection where the user continues straight. Even though the user will most likely continue straight ahead without any additional help, the user might hesitate or would need some kind of confirmation that continuing straight is the right choice. A complex intersection would be a junction where there are at least two adjacent roads of tertiary or secondary class. For this reason, we add confirmation instructions for complex intersections that do not give the user any new information but affirm the user about the correctness of the current path.

The third step connects consecutive decision instructions that are too close to each other. The reason is that when multiple decision instructions are too close to each other, the user will need to prepare for the next maneuver mentally. An example of chained decision instructions is the instruction “Turn left and then turn right”. We have decided to apply this rule only to the decision instructions. The approach instructions are kept intact and refer to only one maneuver at most.

When two consecutive maneuvers form a zigzag pattern, we can improve and simplify the overall maneuver. Instead of telling the user to turn left and then turn right, we can instruct the user to continue straight and describe the intermediate segment (“Continue straight ahead over the bridge”) or describe the type of subsequent road (“Continue slight left onto the cycleway"). These maneuvers are significantly more common on bicycle routes than on car routes. The reason is that cyclists can use the infrastructure designed for cars, pedestrians, and cyclists. The most common case is when the cyclist needs to ride off the pavement onto a road (or the other way), where a simple instruction has to be provided. We apply a zigzag rule to three consecutive edges, where the second segment must be shorter than δ, and the first and third segments must be longer than δ, where the distance δ is set to 10 meters. The merged maneuver is then determined from the sum of the two oriented angles. From this definition, the zigzag maneuver is also applied to maneuvers on the same side (two slight left turns are merged into one left turn). An improvement of our approach would be to consider more than one short segment as part of a longer zigzag maneuver.

Zigzag maneuvers are the only type of compound maneuver that we use. If we define any other compound maneuver, we would need to ensure that the user understands this maneuver (S-turn, loop maneuver, Dubins curves). This problem can be solved by avoiding complex maneuvers when planning routes.

A slightly different approach would be to leave out the maneuvers altogether and use a landmark or other piece of information to guide the user through the route ("Go towards the large church") without requiring the user to follow the route perfectly. In the real world, it does not matter whether the user uses a sidewalk or the adjacent road (or any two parallel paths) as long as the user gets to the destined waypoint on the route. This would solve the previous problem of describing complex maneuvers since we could just direct the user towards some landmark, and we would not care by which way the user reaches this landmark.

The last step in generating the itinerary is related to the verbal announcement of the instructions. We need to make sure that the adjacent instructions do not overlap when executed. For each two adjacent decision nodes (decision instructions), we try to schedule the intermediate instructions so that no two instructions overlap. We call the set of segments between these two decision points a *route leg*. For each individual instruction in a route leg, we create a tuple $I = (D_{start}, D_{processing}, salience)$, where D_{start} is the distance in meters from the start (first decision instruction), $D_{processing}$ is the distance in meters covered during the instruction announcement, and $salience$ expresses the utility of this instruction. The processing distance $D_{processing}$ is estimated from the length of the instruction in words, the user's speed (in km/h) and the reading speed WPM. The words per minute measure for speech is usually in the range of 120 – 160 WPM, we have chosen the conservative value of 130 WPM. The formula is shown in Equation (3).

$$D_{processing} = \frac{speed}{3.6} \times \frac{words}{\frac{WPM}{60}} \tag{3}$$

We also require that each instruction is announced before passing the waypoint (or decision point) to allow the user to process each instruction mentally. The shifted start distance is then expressed as follows:
$D'_{start} = \max(0, D_{start} - D_{processing} - D_{space})$, where D_{space} is an additional space between consecutive instructions. By adding space for each instruction, the user has more time to enjoy the ride and is not bothered by the navigation instructions.

Since we do not need an optimal solution to this planning subproblem, we have opted for a greedy solution. We sort all instruction tuples I by their salience (utility) and plan the instructions from the most salient and check for overlap or duplication of instructions.

The result of all these steps is a route composed of an itinerary with enriched instructions and the underlying path, which may or may not be optimal (depending on whether we used a predetermined path).

Chapter 5
Prototype application

In this chapter, we describe our developed application that implements the approach described in Chapter 4 and show our resulting Web application with a description of its features.

This work uses a PostgreSQL database with a PostGIS extension to represent the data and communicates with a Java application that processes and enriches our routing graph. This application is available through a JSON API and is used for both routing and instruction enrichment. The Web application (client side) was created in such a way that it shows the routes with enriched instructions and enables the user to examine the route and the generated itinerary interactively or to go through the planned route with the help of our application.

The source files of our application are attached to this thesis (content described in Appendix B) or are also available online at `https://gitlab.fel.cvut.cz/michap17/dp`.

Figure 5.1 shows the entire process from the data sources to the Web application.

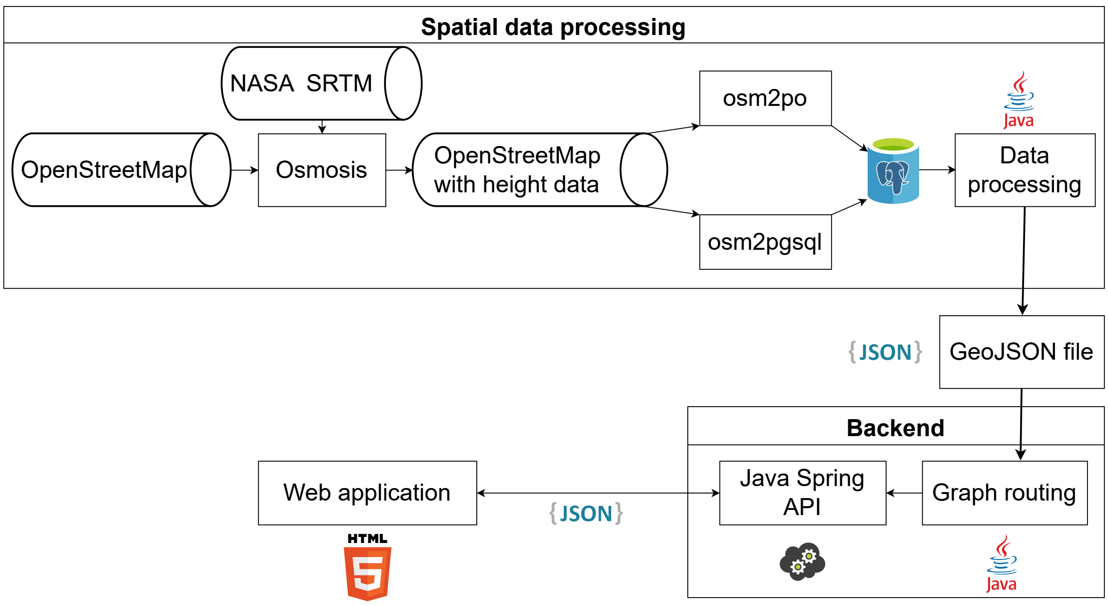


Figure 5.1. Flowchart showing steps from data sources to Web application

5.1 Docker

Both the spatial processing and the backend parts use a range of libraries. We have packaged the whole application into a single Docker image that executes all necessary preparatory steps. This image is created from the Ubuntu Jammy base and installs all dependencies of this application (Java 17 LTS, Maven, PostgreSQL 14, PostGIS, PgRouting, osm2pgsql, and related packages), and configures the database to be populated with data. We assume that the user can download and provide the data from OpenStreetMap and SRTM datasets themselves (we have not included any raw data files in the source files of this project). As our focus was to develop a prototype of an application implementing our approach, we have not aimed our attention at developing a proper software architecture (modularity of the software). We have used a single Maven project that shares the dependencies without any submodules that contains both the part for landmark enrichment of the routing graph and the backend API.

The project can be run completely from the landmark enrichment of the routing graph process to launching the Web application using the `RUN.sh` script. Each part of the project can also be run separately by running the execute goal of the Maven project with the `-Dexec.mainClass` parameter specifying the main class, with the Web server class being the default main class. More details of the software are described in the `README` file in the source files.

If the user is interested only in the Web application and has a processed GeoJSON file (with enriched edges), the project can be used without Docker by building the Java project into a standalone JAR file.

5.2 Backend

This project uses the Spring framework[1] to create a Web server (REST API). Its main purpose is to describe any route using context-enriched instructions and to find a route between any start and end point in our routing graph. Every response of this API is returned in JSON format. The REST API implements these services with parameters supplied using the GET method:

- `/getAllLandmarks` – returns all landmarks present in the dataset as a GeoJSON object (no parameters)
- `/getClosestNode` – finds the closest node for the given coordinates (latitude and longitude), where the response is an index of the closest node (as described in Section 4.4.1)
- `/getRoute` – returns a route and enriched instructions, where the parameters are either:
  a) a pair of start and end indices (obtained using the `getClosestNode` function)
  b) an identificator of a predefined route, which is used for our evaluation
  Optional parameters are the speed (in km/h) and the routing weights (Section 4.4) if the path is not fixed.

The structure of a response to the `getRoute` function is illustrated in Figure 5.2. It contains a list of instructions in plain text format (`text`) and token format (`token`), details of the instruction (`json`), geometries of all referenced landmarks (`landmark`), the geometry of the route, the total length of the route, the complexity of the route as the number of decision points, the estimated time based on the speed of the user, and the total elevation of the route. In the Web application (Section 5.3), the tokens are used to highlight the part of the instructions that refer to any selected landmark shown on the map. The other reason is to detect the names of landmarks that are most likely not in English (in our tested area in Prague), and the Text-to-Speech feature may have problems pronouncing these names.

[1] `https://spring.io/`

```
{
  "instruction": [
    {
      "text": "Go by the ČVUT budova T2 and then continue for 210 m"
      "tokens": [
        {"type": "plain", "text": "Go"},
        {"type": "decision-preposition", "text": "by"},
        ...
      ],
      "json": {
        "type": "Decision",
        "instruction": {
          "landmarks": [
            {
              "type": "building=university",
              "name": "ČVUT budova T2",
              "dist": 31.010887, "angle": 65,
              "sameCategoryLandmarks": 3, "saliency": 0.502865,
              "id": 12220199, "closestLine": {...}
          ]
        },
        "startIdx": 0, "endIdx": 1, "edgeId": 112,
        "priorDecision": 0.25, "length": 207.169298,
        "elevation": 1.65,
        "properties": {
          "steepness": 0, "_osm": 4044403, "name": "Technická",
          "highway": "residential", "class": 41, "oneway": 1
        },
        "distanceCovered": 23.076923
      },
    }
    ...
  ],
  "landmark": {
    "12220199": {
      "type": "Feature",
      "geometry": {...},
      "properties": {...}
    },
    ...
  }
  "time": 61.962563,
  "timeString": "2 minutes",
  "elevation": 9,
  "geometry": {...},
  "numDecisionPoints": 2,
  "length": "260 m",
}
```


Figure 5.2. API response of the `getRoute` function

5.3 Web application

The functions of the backend layer are utilized in our Web application, which presents the routes in a human-friendly way. It uses the Mapbox GL JS[2] library to visualize data on a map, the Water.css[3] CSS style (with builtin dark mode), the SVG icons from `http://svgicons.sparkk.fr/` and the Pace[4] script that shows a progress indicator during page loading.

The Web application page is divided into two areas, visible in Figure 5.3. The right area is occupied by the map showing the route and its landmarks. The left area has a panel with the search form, the itinerary with instructions, and the settings panel to alter the behavior of the application (details below).

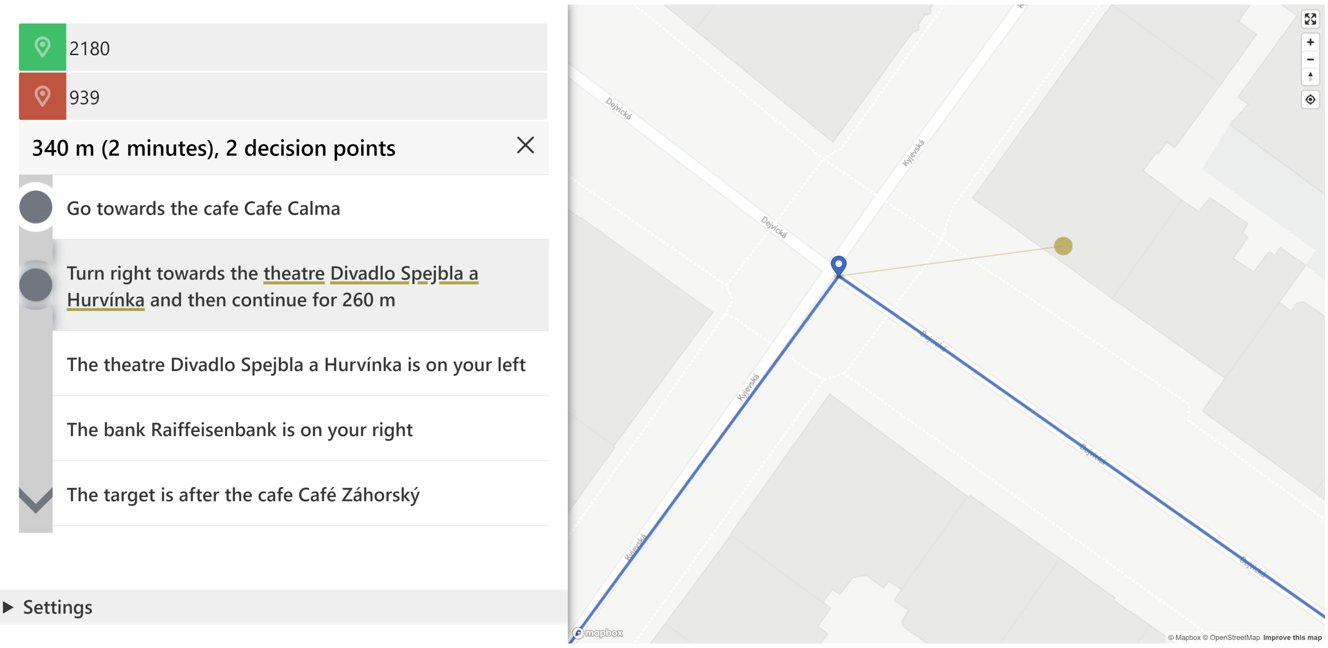


Figure 5.3. Screenshot of the Web application with a route and a selected instruction

The application was built to be mobile-friendly (uses a responsive design) and includes both a light and dark design (selected based on the user preference), as shown in Figure 5.4. The Web application is available online at `https://michap17.cz.eu.org`.

5.3.1 Route visualization

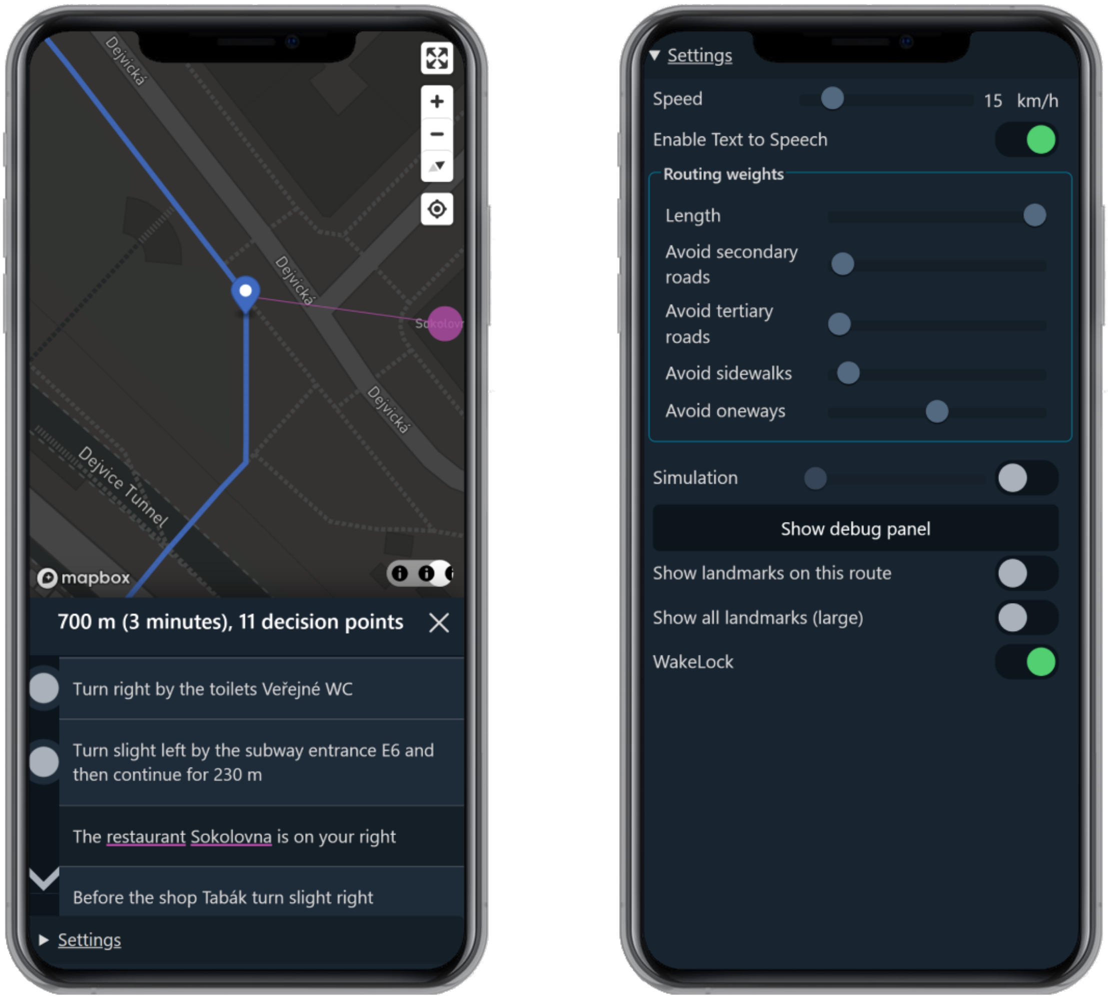
Initially, the user needs to search for some route. This can be accomplished in four ways. The user can either right-click on the map and select the start or end point using the context menu, focus either the start or end field in the search form and then click inside the map to select the start or end point, enable geolocation to select the start point closest to the user, or visit the site using an URL with preset start and end point parameters. After that, the route is shown on the map and the itinerary with instructions is shown on the left side.

After a route is found, the map is used to show the geometry of the route and the left panel is filled with corresponding instructions. The left panel also contains the

[2] `https://www.mapbox.com/mapbox-gljs`
[3] `https://watercss.netlify.app/`
[4] `https://codebyzach.github.io/pace/`

length of the route in meters, the estimated time it will take to complete the route, and the complexity of the route expressed in the number of decision points (also visible in Figure 5.3). The user can then select any instruction to show the position on the route and the landmarks referenced by this instruction. Each type of instruction is differentiated using an icon, and each landmark is differentiated by a unique color. The user can also select any landmark on the map to see its properties in the right panel.

The right panel is initially hidden and it can be shown in two ways. The user can either click on any landmark on the map or manually open the right panel using a button in the settings. When the right panel with details is shown and the user selects any instructions, the right panel will be filled with details of this instruction.



Navigation Settings

Figure 5.4. Screenshot of the Web application in dark mode on a mobile device

5.3.2 Settings panel

The user can open the panel with settings to change the routing properties, show all landmarks used on this route or the whole dataset, show the right panel with details, enable the Text-to-Speech function, or run a simulation of the route traversal (shown in

Figure 5.4[5]). The Text-to-Speech uses the Speech Synthesis browser API[6] which uses the voices available in the system. The user must, in some rare cases (depending on the device), enable the Text-to-Speech functionality and install a voice package manually. Instructions are read aloud using any available voice and names (of landmarks and streets) are preferably read in Czech.

The settings panel also contains a section with routing parameters. Users can modify the weight of the length of the route or the weights for secondary and tertiary roads, one-way streets, or pedestrian ways. The reason for this feature is that users have varying preferences and requirements for their optimal route. Each route can be iteratively refined by adjusting these parameters to find a route that the user desires. The last parameter is the speed of the user. This parameter is used during the itinerary generation (Section 4.7) and it influences the density of the instructions.

5.3.3 Route traversal

An additional feature of this application is to examine the way in which instructions are announced throughout the route. The user can then explore the route using an animation that goes through the route using the set speed from the start to the end and highlights the instructions at each step of the route. This animation is a simplified simulation of the route traversal and does not take into account factors such as the change of speed at the intersections and when going uphill or downhill, or the inaccuracy of the GPS. If the user has the Text-to-Speech functionality enabled, the application will read these instructions aloud.

The user can also enable geolocation, which tracks the user position and checks if the user has passed any instruction waypoint. The behavior is similar to the route animation, with the only difference being the tracking of the user's position and heading, as in the animation the position was snapped to the path, and the heading was estimated from the path. Although this application is only a prototype and not a production-ready application, it can be used as a navigation application in a real environment.

[5] Phone mockup image from `https://unsplash.com/photos/ZOXwrpmYWrA`

[6] `https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis`

Chapter 6
Evaluation

We have decided to evaluate our approach on a set of 10 different routes in the area of Prague 6, where we have manually defined the set of landmarks that must be mentioned so that the user can successfully complete the route. In addition, we have picked out three of these routes that we examine in greater detail by analyzing the correctness of the use of landmarks in the generated instructions. The remaining routes are only checked for the presence of predefined landmarks without checking the context of these landmarks. The set of manually defined landmarks is based on map data, panorama images (Seznam.cz Panorama), and personal knowledge of the area. Routes have been chosen so that they would contain shorter and longer routes, both in the city and in nature, crossing or bypassing different landmarks and leading through different types of roads. The following section is structured by describing the route, the associated landmarks, and the performance of our algorithm on these test cases.

All these routes are shown in Appendix A with map images and itineraries of the generated instructions.

6.1 In-depth analyzed routes

In this set of routes, we not only check the presence of the landmark in the itinerary, but we also analyze how this landmark is used. The landmark can be used improperly if it is not visible in the real world or the landmark is used with an incorrect preposition (leading the user away from some landmark instead of towards it).

6.1.1 Route 1

Landmark	OpenStreetMap IDs	Present
Gymnázium Arabská	8308536	✔
police station	22635267, 22635266	✘
Sídliště Červený Vrch stop	4719188957, 180465704, 55538877, 548761682, 2437837	✘
Červený Vrch stop	1909037962, 180465716, 180465710, 3414583439, 2415016	✔
Základní škola a mateřská škola Červený vrch	8267963	✔
ZŠ a MŠ Na Dlouhém lánu	371733348	✘
railroad tracks / crossing	8085377, 282736943	✘
Zahradnictví Chládek	318003493, 65639051	✔

Table 6.1. Landmarks for route A.1

This route (shown in A.1) combines map areas with a dense and sparse number of landmarks and contains a wide range of road types, ranging from paths to cycleways. It also contains a few densely packed decision points where the user has to change the orientation in rapid succession. The landmarks visible on the route are mainly public transport stops or public amenities (school, police station). A police station is a salient landmark, but in this case, the police station consists of two buildings (viewed as two separate landmarks of the same category), and consequently, the uniqueness of both stations decreases. The other reason is that information about the nearby school and about the consequent decision instruction is dominant.

Gymnázium Arabská is mentioned twice at the beginning of the route. It is used in a decision instruction ("Turn right towards the school **Gymnázium Arabská** and then continue for 190 m") and in the following confirmation instruction ("The school **Gymnázium Arabská** is on your left"). In both cases, the landmark is used correctly and should be helpful for this route.

Červený Vrch stop is mentioned three times, making it one of the key landmarks in this route. It is first mentioned in a confirmation instruction ("The tram stop **Červený Vrch** is on your left"), then it is used later in two subsequent decision instructions ("Turn right by the tram stop **Červený Vrch** and then turn left" and "Turn left away from the tram stop **Červený Vrch** and then continue for 520 m"). The last decision instruction may be surprising to the user for the reason that the preposition *away from* is ambiguous, putting the user at risk of deviating from the route. If this second decision instruction did not contain a maneuver, the user could continue away from the stop in the wrong direction.

The school **Základní škola a mateřská škola Červený vrch** is mentioned only as confirmation and with the correct relative side ("The school **Základní škola a mateřská škola Červený vrch** is on your left"). The school *ZŠ a MŠ Na Dlouhém lánu* is not mentioned because of the lower salience compared to the river *Dejvický potok*.

The **Zahradnictví Chládek** garden center is a landmark that is mentioned twice on this route. First, it is mentioned in a decision instruction ("Turn right towards the shop **Zahradnictví Chládek** and then continue for 160 m") and then in the second to last approach instruction ("The target is after the shop **Zahradnictví Chládek**"). The decision instruction could have used the information about the railroad crossing that is closer and visible from the oncoming direction, because the garden center is not visible at the point where the instruction is announced, making the use of this landmark ineffective.

The landmarks on this route have been used correctly in most cases. The main problem was not mentioning some landmarks that we had expected to be included in the itinerary.

6.1.2 Route 2

Route A.2 goes from a larger public transport station Nádraží Veleslavín towards the Džbán water reservoir. Because the user starts at the station, the station is not mentioned, as it does not provide any meaningful information to the user. The only use we could imagine would be to tell the user to go away from it. This is not used because there are other more useful landmarks around this place. The fuel station is visible at the beginning of the route, but it is located more than 50 meters from the path, so it is not even considered as a landmark.

The sports center **SK Aritma Praha** is a large area that follows a considerable part of this route. This landmark is first introduced by an approach instruction ("Before the

Landmark	OpenStreetMap IDs	Present
Nádraží Veleslavín stop	3688348028, 3414583403, 3449320305, 5316448454, 5155737183	✗
fuel station MOL	9109958351	✗
ZŠ Vokovice	8268336	✗
playground	2578636964	✗
SK Aritma	3732517330, 318015760	✓
camp Džbán	365359592	✗
parking	29985104	✓
Džbán	5431533	✓

Table 6.2. Landmarks for route A.2

sports center **SK Aritma Praha** continue straight") that precedes a decision instruction ("Continue straight by the sports center **SK Aritma Praha** and then continue for 540 m"). Later, it is mentioned on the part of the route when this landmark is no longer visible ("After the sports centre **SK Aritma Praha** turn right"), even though there are more relevant landmarks (a water stream or the water reservoir Džbán). The cause of this is that the landmark is evaluated to be visible and the category of this landmark is set too high (sports centers have a weight of 0.70, as shown in Table 4.1).

The **parking** area is part of the route in the Džbán area and is used only in an approach instruction ("After a **parking** turn left"). This instruction is used with the correct preposition and at the correct point of the route.

The water reservoir **Džbán** is near the end of the route, but due to its size, it may not be useful for pinpointing exact points on the route (such as the end of the route). It is used only once on the route as a confirmation ("The reservoir vodní nádrž **Džbán** is on your right"), but in this case the name of the water reservoir may be irrelevant information to the user.

Although some landmarks were not mentioned at the beginning of the route, the landmarks included in the itinerary were used correctly.

6.1.3 Route 3

Landmark	OpenStreetMap IDs	Present
Lotyšská stop	3413902181, 3413902182, 153528609, 153528613	✗
CIIRC	405845545	✓
Pizzeria Grosseto	282690498	✗
Cyklo 69	756569272	✗
Vítězné náměstí stop	55552358, 5686152363, 4719188956, 4518007395, 2454089	✓
Kafkova stop	1250854420	✗

Table 6.3. Landmarks for route A.3

Route A.3 leads through an area that is easily passable for cars, but cyclists may have trouble or feel at risk due to this area of the city being designed mainly for cars. This may pose a challenge to the routing algorithm, as the cyclist needs to use the car infrastructure or take a long detour (both unfavorable outcomes). Alternative routes may lead through some minor roads or through an underpass. This area also includes multiple public transport stops that are useful for orientation purposes. The Lotyšská stop is not mentioned because the start location is next to it and cannot be used to guide the user along the route. This case is later mentioned in the next section discussing the results.

The building Czech Institute of Informatics, Robotics and Cybernetics (CIIRC, in Czech **Český institut informatiky, kybernetiky a robotiky**) is used only once at the beginning of the route ("Continue straight away from the **Český institut informatiky, robotiky a kybernetiky** and then continue onto Jugoslávských partyzánů Street"). This landmark is used in the compound zigzag maneuver. If there was no operation to merge two consecutive zigzag maneuvers, there would be two decision instructions: "Turn left away from the **Český institut informatiky, robotiky a kybernetiky**" and "Turn right onto Jugoslávských partyzánů Street". Although the user goes away from this building for a short time, the route then continues along this building. For this reason, we classify this instruction as potentially misleading. The second problem with this landmark is the length of its name, as our aim was to create short and concise instructions.

The **Vítězné náměstí** stop is used only once as a confirmation ("The tram stop **Vítězné náměstí** is in front of you"). Despite the fact that this instruction is used correctly, we expected the landmark to be used as part of a decision instruction (e.g., "Continue towards the tram stop **Vítězné náměstí**") and provide more use for this route.

This route was different from the first two by going through an area with an abundance of landmarks and graph features. The difficulty of this route is thus to select the appropriate information for the user.

6.2 Additional routes

In this section, we evaluate the remaining routes only by checking the presence of the defined landmarks in any instruction. We then comment on the reasons why some landmarks were not included in the itinerary and analyze the overall results of our application for each route.

Landmark	OpenStreetMap IDs	Present
FEL ČVUT	12220199	✔
NTK	28090508	✔
FA	405845543, 100959765	✔
FSv	9566872590, 99013221, 1038858634	✘
Studentský dům	25550769	✔

Table 6.4. Landmarks for route A.4

Route A.4 leads from the Faculty of Electrical Engineering building of the Czech Technical University (in Czech: FEL ČVUT) to the Studentský dům building. The

Landmark	OpenStreetMap IDs	Present
pond Srpeček	9204470	✗
railway bridge	981696630, 239776758, 25738285	✗
bridge Císařský ostrov	13331942	✓
Jezdecká hala	95350189	✓
Trojská lávka	12221515	✓
Občerstvení U lávky	3448110894	✓
Trojský zámek	23607662	✗

Table 6.5. Landmarks for route A.5

route leads through the campus with other universities (FA, FSV) and by the large oval building of the National Library of Technology (NTK).

Route A.5 examines the performance of our method in a natural area. The area starts in Stromovka Park and leads over multiple bridges to the Trojský zámek (Troja Palace). The Trojský zámek landmark is not mentioned in the itinerary since there is an artwork in front of it that is considered more prominent. Unfortunately, the real situation is most likely to be the opposite.

Landmark	OpenStreetMap IDs	Present
ÚTVS	25552026	✓
Stanice techniků	30147457	✓
Armáda ČR	1000625667	✗
Restaurace Pod Juliskou	332218800	✓
Nádraží Podbaba stop	1662077479, 8157899805, 153528625, 153528630, 3413709752	✓

Table 6.6. Landmarks for route A.6

Route A.6 is a shorter route between the Institute of Physical Education and Sport at CTU (in Czech: ÚTVS ČVUT) and the Nádraží Podbaba public transport station. This area contains many possible landmarks. Some not included landmarks are the tram rails, the secondary road Podbabská, or the Vlaštovka kindergarten, which resides in the same building as the Stanice techniků.

Landmark	OpenStreetMap IDs	Present
bridge	25401460	✓
Kramářova vila	61973096	✗
cafe Hanavský pavilon	28701169	✓
Stalinův pomník / Metronom	5027320564, 12273886	✗
Letenský zámeček	25401471	✗

Table 6.7. Landmarks for route A.7

Route A.7 starts at the edge of Chotkovy sady and ends at Letenské sady. In both parks, there are only a few landmarks, where some of them are not salient and unique enough (benches, trees), and there is greater visibility, which may cause some landmarks to be missed using our approach. The Stalinův pomník landmark is missed because there is a wall (way 455283557) between the path and the landmark. This wall separates the way from the heightened area, where the landmark is located, and therefore this landmark is visible in the real world. The Letenský zámeček landmark is visible, but there is a closer biergarten (beer garden, relation 4469371) that is selected, and therefore the expected landmark is not mentioned throughout the route.

Landmark	OpenStreetMap IDs	Present
playground	7653685	✓
Dejvický potok	134664205	✓
ZŠ a MŠ Na Dlouhém lánu	371733348	✗
Stadion Hvězda	7652745	✓
SK Střešovice	32108335, 32108334, 371733345	✓
Pod Vyhlídkou stop	358463677, 358463642	✓
Kino Ořechovka	2571994	✓
Dělostřelecká stop	5724247013	✗
Klubovna	2338683459	✓

Table 6.8. Landmarks for route A.8

Route A.8 is a slightly straighter route (with only 11 decision points) that leads from the Veleslavín area to the Kafkova street. Throughout the route, there are a variety of types of landmarks. The only missed landmarks are the school ZŠ a MŠ Na Dlouhém lánu (for the same reason as in Route A.1) and the Dělostřelecká public transport stop. The reason for missing the stop is that a public transport station in the opposite direction (node 5724247004) is later used for the approach instruction, and this previous landmark is discarded as a duplicate landmark.

Landmark	OpenStreetMap IDs	Present
Špitálka stop	2427455, 92049293, 92049291	✗
power substation	30132249	✗
Šárecký potok	10619589, 26377899, 32114840, 449447618, 684849446, 684849447, 684849448, 26377906, 959123898, 1011684530, 1011684529, 959123897	✓
bridge	61007950	✗
Kalinův mlýn stop	390377563, 390377562	✓
Kuliška stop	390377561, 390377560	✓
Korek stop	390377558, 390377557	✓
restaurant Jenerálka	8811002917	✓

Table 6.9. Landmarks for route A.9

Route A.9 leads through the natural area of Divoká Šárka that contains many natural landmarks (waterways, greenery) and a few bus stops. Another benefit of this route is its simplicity, as it contains only eight decision points. Therefore most of these landmarks are mentioned in the form of confirmation instruction. Some bridges are not mentioned as they are part of the route, so the instruction does not refer to them as a landmark, but just as road information ("Continue over the bridge").

Landmark	OpenStreetMap IDs	Present
artwork	6909296641	✓
playground	1018486001	✗
Maroldovo panorama	24399672	✓
Planetárium	22634016	✓

Table 6.10. Landmarks for route A.10

The last route A.10 is focused entirely on confirmation instructions in an environment without any urban landmarks (shops, restaurants, stops). As this route leads through Stromovka Park, there are only a limited number of landmarks, so the selection of a landmark is significantly easier. Sometimes there is only one landmark, so the choice of the most salient landmark is trivial in these cases.

6.3 Discussion

The results presented in the previous section should provide an idea of the advantages and shortcomings of our work. A limitation of our work is that the results are heavily influenced by the map data. The results may differ significantly for other map areas due to the varying levels of map coverage in the OpenStreetMap data.

We would like to address the reason why some landmarks were not mentioned. The first possible situation is that the landmark was too far away or there was some obstacle between the route and the landmark. The second possibility is that there was another landmark that was deemed more salient. The third reason is that the instruction referring to this landmark would overlap with another more valuable instruction, so this instruction is not used in the itinerary. The next reason is that this landmark would not provide the user with any meaningful information. An example of this case is the Lotyšská tram stop route shown in Table 6.3. In this case, the landmark cannot be used to provide any useful information on the direction in which the user should go. The last reason is that there was some information on the road that we considered more important (bridge, stairs).

To give an idea of the amount of additional information we can provide the user, we have compiled the statistics in Table 6.11. The row *Test cases success rate* is a number of predetermined landmarks that were present in the 10 test cases (mentioned above). The possible reasons for missing some landmarks in the test cases have already been addressed above. As we can see, the majority (86.35%) of the decision points contain at least one nearby landmark. The predominant road information is the information about the class of the road (as expected), since this type of information is almost always present in OpenStreetMap. The number of junctions with a distinct shape is also a notable result (one in three junctions), with the four-way junctions being the most common shape.

This work does not include user experiments to determine the real impact of our method, as we have evaluated our method only on a set of 10 test routes. Such a user experiment could be conducted in a real environment or with panorama images (Google Street View, Seznam Panorama) to carry out the experiment digitally, which could verify (or refute) our claims. The source of errors that we have ignored is human error (misinterpretation of the instruction), the change of environment (name of the store changes, the landmark is partially obstructed), or the instruction being misleading. An instruction can be misleading when there are multiple landmarks of the same category, which can cause the user to turn at the wrong intersection ("Turn left at the second fountain", in [11] described as multiple landmarks on the same route leg), or make a wrong turn ("Go towards the bench" at a junction with multiple benches), or the landmark causes confusion for the user ("After the church turn right towards the church").

Data	Absolute value	Relative value
Test cases success rate	38 / 65	58.46 %
Decision points with any landmark	22474 / 26028	86.35 %
Decision points with road change	19770 / 80148	24.67 %
Decision points with any road information	19324 / 19770	97.74 %
Bridge	169 / 19770	0.85 %
Stairs	960 / 19770	4.86 %
Crosswalk	2281 / 19770	11.54 %
Sidewalk	805 / 19770	4.07 %
Hill	674 / 19770	3.41 %
Road class	14435 / 19770	73.01 %
Junctions with distinct shape	9207 / 26028	35.37 %
T junctions	2966 / 26028	11.40 %
Y junctions	633 / 26028	2.43 %
Four-way junctions	5608 / 26028	21.55 %
Confirmation instructions with any landmark	11506 / 26028	44.21 %
Edges with best continuation	19824 / 26028	76.16 %

Table 6.11. Statistics of map data

An aspect of our work that is not evaluated is the landmark naming strategy. Each landmark is given a name based on its category and name. We have not constructed any complex naming method, which can often lead to an excessively long description. The first problem with this method is that some categories of landmarks do not need a name because their category is significant on its own. Examples of these landmarks can be churches, bridges, chimneys, statues, castles, or any other well-known types of objects.

The second problem is related to the length of the landmark's name. Some landmarks have remarkably long descriptions, such as *Ústav tělesné výchovy a sportu Českého vysokého učení technického* (way 25552026) or *Dejvická kolej - správa účelových zařízení*

ČVUT v Praze (way 26476984). A partial solution would be to use a shorter name (key `short_name`), which unfortunately is not always specified.

Other minor problem is the fact that most of the names in the map data are in Czech, and some of them include the category in their name. This can lead to the so-called *bilingual tautological expressions*[1], where we combine words that mean the same thing in different languages. An example of this effect is the hotel *Hotel International Prague* (way 25550853), the cemetery *Hřbitov Bubeneč* (way 38066853), or the school *Základní škola a mateřská škola Na Dlouhém lánu* (way 371733348).

The names of landmarks could also be improved by including not only their category and name. Some landmarks are unique due to their color ("turn right at the yellow house"), height (skyscrapers), material, shape, or any other unique aspect of the landmark. Our conservative approach is partly caused by the limits of the OpenStreetMap data, as the color or height of buildings is only rarely specified.

We conclude that, on the basis of the results, our method is a viable and meaningful method for enriching navigation instructions. Although these enriched instructions may be considered redundant, in bicycle navigation, this can help reduce the cognitive load or minimize the risk of route deviation due to human error.

[1] `https://en.wikipedia.org/wiki/Pleonasm#Bilingual_tautological_expressions`

Chapter 7
Conclusion

This thesis is a continuation of the bachelor's thesis [26] and builds on its findings to generate context-enriched navigation instructions. We have categorized the navigation instructions into three types. The most important instruction is the decision instruction with a maneuver that tells the user to change the direction of travel. The approach instruction prepares the user for an oncoming decision instruction by providing details of the location where the decision will be taken. The last is the confirmation instruction that reassures the user about the correctness of the path at the edges of the route by describing the environment of the route.

Each instruction can refer to some landmark (salient object) or some property of the road (class of the road, stairs, bridge) or the shape of the junction (T junction, Y junction, four-way junction). We have used a set of operations to transform the instructions and integrate them into a route itinerary. These operations include removing irrelevant and duplicate instructions, describing zigzag maneuvers, linking consecutive decision maneuvers, and detecting instruction overlap (based on the verbal announcement). Each instruction with a landmark also contains the position of the landmark relative to the path. If the landmark overlaps the path, we also detect the height relation (a bridge can go over the path or the path can go over a bridge). We have also shown possible difficulties in detecting the spatial relationship between nested landmarks (a point landmark inside a building). We have used OpenStreetMap data to extract both the routing graph and the landmarks, thus some methods in this work may be specific to this dataset.

Finally, we have implemented this approach in a Java application. Our approach has been demonstrated using a Web application available at `https://michap17.cz.eu.org`. The results of our work have been evaluated on a set of test routes, and the results and properties of our algorithm have been discussed. Although choosing the correct context information may be a subjective matter, we have demonstrated that using OpenStreetMap data is a viable option not only as a source of a routing graph but also for extracting landmarks and features of the routing graph, which can then be incorporated in context-enriched navigation instruction. In bicycle navigation, descriptive and context-enriched navigation instructions allow the user to understand the environment more easily and execute the required navigation instructions accurately.

7.1 Future Work

As described in the Evaluation (Chapter 6), our solution is not robust enough and tested to replace traditional navigation services. We recommend integrating our method of enriching instructions into an already-made (bicycle) navigation system. There already exist a large number of ready-made general route services (Google Maps, Mapy.cz), bicycle route planners (Cyclers.tech) or routing engines (osm2po, OSRM, GraphHopper, OpenRouteService, ...)[1]. If our approach was integrated into an already-made nav-

[1] `https://wiki.openstreetmap.org/wiki/Routing`

igation system, the routing parameters (simplest route, avoid busy roads, sidewalks, one-ways, and total length) could be used as criteria (weights) in the planner, and the method for generating an itinerary of enriched instructions would remain unchanged.

Another improvement is the salience formula (described in Section 4.1.7). We have designed a simple salience formula, but salience in the real environment is much more complex and hard to estimate. An extension would be to use machine learning to calculate the weights of each criterion since we assigned the weights based on expert knowledge from the literature ([7–8, 13]). This work can be extended by using information from datasets other than OpenStreetMap. Images from social networks or text resources (news articles, Wikipedia) could be used to determine the popularity of some objects, which is likely to be correlated with their salience.

Another extension of this work is related to the grouping of (aggregating) landmarks. When there is a group of benches, trees, or buildings (housing area, university building complex, block of buildings), we see them as individual landmarks rather than as a group. This extension would require a technique to detect when separate disjoint objects form a group and when not.

We may also define other types of instructions. The user can follow some linear landmark (“Follow the railroad tracks”), avoid a specific landmark (“At the next intersection, continue along the way without the hedge fence”), or we can use counting for the approach instruction (“At the third intersection turn left”). In this work, we have avoided counting as a type of instruction to prevent any ambiguity related to the term intersection.

Other types of instruction that we have not used (mentioned in [1]) are the description of landmarks and commentaries. The description of landmarks might be useful in cases where the category and the name of the landmark are not sufficient and the landmark is a vital landmark for the route. Commentaries (“It will not take long” or “The walk requires 15 minutes”) may be useful in some situations without any landmarks, but we perceive their usefulness as very low. The other piece of information that could be provided is an overview of the plan at the beginning of the route (“This route leads through Stromovka Park and will take you approximately 10 minutes”).

In summary, this thesis should serve as a springboard for future research on more human-friendly navigation instructions.

References

[1] Michel Denis. *The description of routes: A cognitive approach to the production of spatial discourse.* In: ADRSC, 1997. 409-458.
https://www.researchgate.net/publication/242164650_The_description_of_routes_A_cognitive_approach_to_the_production_of_spatial_discourse.

[2] Adam Rousell, and Alexander Zipf. Towards a Landmark-Based Pedestrian Navigation Service Using OSM Data. *ISPRS International Journal of Geo-Information.* 2017, 6 (3), DOI 10.3390/ijgi6030064.

[3] Markus Dräger, and Alexander Koller. *Generation of landmark-based navigation instructions from open-source data.* In: Avignon, France: Association for Computational Linguistics, 2012. 757–766.
https://www.aclweb.org/anthology/E12-1077.pdf.

[4] Jakub Krukar, Angela Schwering, and Vanessa Joy Anacta. Landmark-Based Navigation in Cognitive Systems. *KI - Künstliche Intelligenz.* 2017, 31 (2), 121-124. DOI 10.1007/s13218-017-0487-7.

[5] Andrew J. May, Tracy Ross, Steven H. Bayer, and Mikko J. Tarkiainen. Pedestrian navigation aids. *Personal and Ubiquitous Computing.* 2003-12-1, 7 (6), 331-338. DOI 10.1007/s00779-003-0248-5.

[6] Kristin L. Lovelace, Mary Hegarty, and Daniel R. Montello. *Elements of Good Route Directions in Familiar and Unfamiliar Environments.* In: Berlin, Heidelberg: Springer Berlin Heidelberg, 1999-12-16. 65-82. ISBN 978-3-540-66365-2.
http://link.springer.com/10.1007/3-540-48384-5_5.

[7] Martin Raubal, and Stephan Winter. *Enriching Wayfinding Instructions with Local Landmarks.* 2002.
http://link.springer.com/10.1007/3-540-45799-2_17.

[8] Alexander Klippel, and Stephan Winter. *Structural Salience of Landmarks for Route Directions.* In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. 347-362. ISBN 978-3-540-28964-7.
http://link.springer.com/10.1007/11556114_22.

[9] Ariane Tom, and Michel Denis. Language and spatial cognition. *Applied Cognitive Psychology.* 2004-12-15, 18 (9), 1213-1230. DOI 10.1002/acp.1045.

[10] Monika Sester, and Sagi Dalyot. *Enriching Navigation Instructions to Support the Formation of Mental Maps.* In: Cham: Springer International Publishing, 2015. 15-33. ISBN 978-3-319-19949-8.
http://link.springer.com/10.1007/978-3-319-19950-4_2.

[11] Matt Duckham, Stephan Winter, and Michelle Robinson. Including landmarks in routing instructions. *Journal of Location Based Services.* 2010, 4 (1), 28-52. DOI 10.1080/17489721003785602.

[12] Alexander Klippel, Stefan Hansen, Kai-Florian Richter, and Stephan Winter. Urban granularities—a data structure for cognitively ergonomic route directions. *GeoInformatica*. 2009, 13 (2), 223-247. DOI 10.1007/s10707-008-0051-6.

[13] Anita Graser. Towards landmark-based instructions for pedestrian navigation systems using OpenStreetMap. *Societal Geo-Innovation: short papers, posters and poster abstracts of the 20th AGILE Conference on Geographic Information Science. Wageningen University & Research*. May 2017, 9-12.

[14] Adam Rousell, Stefan Hahmann, Mohamed Bakillah, and Amin Mobasheri. Extraction of landmarks from OpenStreetMap for use in navigational instructions. *Proceedings of the AGILE Conference on Geographic Information Science, Lisbon, Portugal*. 9-12.

[15] Heinrich Löwen, and Angela Schwering. *An algorithm for the selection of route dependent orientation information*. 2019.

[16] Matt Duckham, and Lars Kulik. *"Simplest" Paths.* In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. 169-185. ISBN 978-3-540-20148-9.
http://link.springer.com/10.1007/978-3-540-39923-0_12.

[17] Birgit Elias. *Extracting Landmarks with Data Mining Methods.* In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. 375-389. ISBN 978-3-540-20148-9.
http://link.springer.com/10.1007/978-3-540-39923-0_25.

[18] Antonia Golab, Markus Kattenbeck, Georgios Sarlas, and Ioannis Giannopoulos. It's also about timing! When do pedestrians want to receive navigation instructions. 1-33. DOI 10.1080/13875868.2021.1942474.

[19] Jukka M. Krisp. Landmarks for Location-Based Services (LBS) in Particular Navigation and Wayfinding. *KI - Künstliche Intelligenz*. 2017, 31 (2), 199-201. DOI 10.1007/s13218-016-0462-8.

[20] *Mitsubishi Electric Develops Scene-Aware Interaction Technology*. 2020.
https://emea.mitsubishielectric.com/en/news/releases/global/2020/0722-a/index.html.

[21] *At Last, a Self-Driving Car That Can Explain Itself*. 2022.
https://spectrum.ieee.org/at-last-a-self-driving-car-that-can-explain-itself.

[22] Wilko Heuten, Niels Henze, Susanne Boll, and Martin Pielot. Tactile wayfinder. *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*. 2008, 172-. DOI 10.1145/1463160.1463179.

[23] Jan Balata, Zdeněk Míkovec, Petr Bureš, and Eva Mulíčková. Automatically Generated Landmark-enhanced Navigation Instructions for Blind Pedestrians. DOI 10.15439/2016F135.

[24] Jan Balata, Zdeněk Míkovec, and Pavel Slavík. *Landmark-enhanced route itineraries for navigation of blind pedestrians in urban environment.* In: 2018. 181-198.
http://link.springer.com/10.1007/s12193-018-0263-5.

[25] Jan Balata, Zdeněk Míkovec, and Ivo Malý. *Navigation Problems in Blind-to-Blind Pedestrians Tele-assistance Navigation.* In: *Proceedgings of Human-Computer Interaction - INTERACT 2015*. 2015. 89–109. ISBN 978-3-319-22700-9.
http://link.springer.com/10.1007/978-3-319-22701-6_8.

[26] Petr Michalík. *Obohacování grafů dopravních sítí o kontextové informace z OpenStreetMaps*. 2020-06-08.
http://hdl.handle.net/10467/87672.

[27] Marie Demlová. *Text přednášek z letního semestru 2017/18 v jednom souboru*.
https://math.fel.cvut.cz/en/people/demlova/lgr/text_lgr_2018.pdf.

[28] *Definition:Graph (Graph Theory)*. 2008-.
https://proofwiki.org/wiki/Definition:Graph_(Graph_Theory).

[29] Pavol Žilecký. *Junction-aware Multicriteria Bicycle Route Planning*. 2015.
http://hdl.handle.net/10467/61841.

Appendix A
Itineraries of the test routes

Itinerary icons

Approach instruction
Confirmation instruction
Decision instruction

Start
End

Figure A.1. Route 1

A.1 Route 1: Vokovice – Zahradnictví Chládek

Go east and then continue for 190 m
At a gate turn right
Turn right towards the school Gymnázium Arabská and then continue for 190 m
The school Gymnázium Arabská is on your left
Before the restaurant Pizzeria Slamini continue straight
A bench is on your left
Continue straight towards the restaurant Pizzeria Slamini
Turn right away from the restaurant Pizzeria Slamini and then continue for 290 m
A power substation is on your left
Before a post box continue straight
The tram rails is on your left
Continue straight onto the sidewalk
Continue straight by the parcelpickup vending machine AlzaBox and then continue for 180 m
The tram stop Červený Vrch is on your left
Turn left over the crosswalk
Turn right by the tram stop Červený Vrch and then turn left
Turn left away from the tram stop Červený Vrch and then continue for 520 m
The school Základní škola speciální a praktická škola Rooseveltova is on your right
The school Základní škola a mateřská škola Červený vrch is on your left
The industrial area Teplárna Veleslavín is on your right
A playground is in front of you
After a gate continue straight
Continue straight towards the stream Dejvický potok
Turn slight left onto the path and then continue for 390 m
The stream Dejvický potok is on your right
At the T junction turn right
Turn right by the stadium HC Hvězda Praha and then continue away from a bollard
Continue straight away from a bollard
Turn right towards the shop Zahradnictví Chládek and then continue for 160 m
The target is after the shop Zahradnictví Chládek
A gate is on your right
The target is in front of you

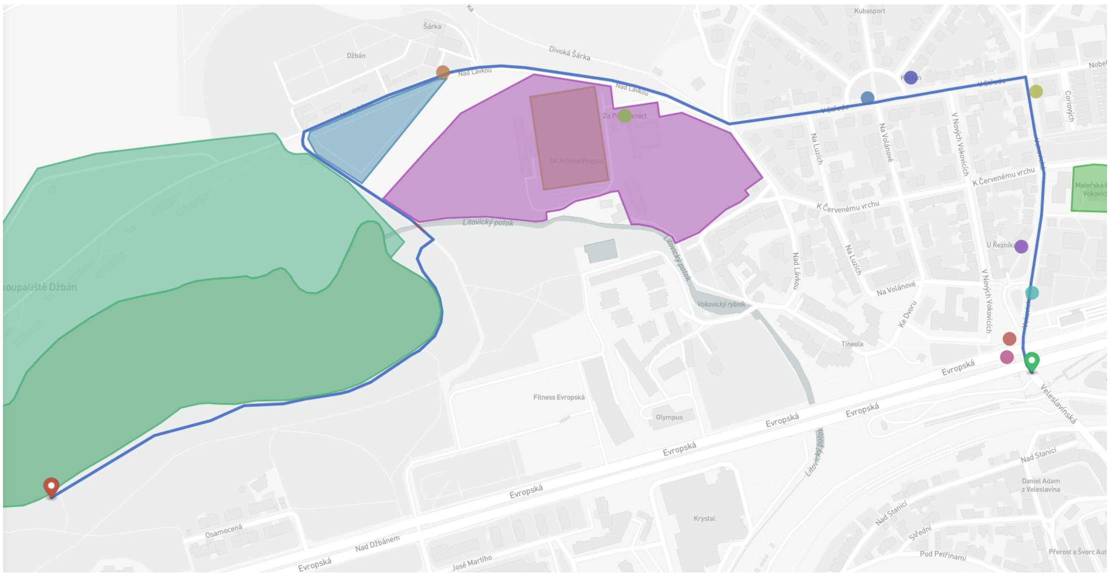


Figure A.2. Route 2

A.2 Route 2: Veleslavín – Džbán

Go towards the subway entrance E4 and then continue for 370 m
The subway entrance E2 is on your left
The public transport platform Nádraží Veleslavín is on your right
The shop Václav Caithaml is on your left
The kindergarten Mateřská škola Vokovická is on your right
After a recycling container turn left
Turn left by a recycling container and then continue for 370 m
The restaurant Kaštan is on your right
A parking ticket machine is on your right
Before the sports centre SK Aritma Praha continue straight
Continue straight by the sports centre SK Aritma Praha and then continue for 540 m
The restaurant Za Pět dvanáct is on your left
A pitch is on your left
A gate is on your right
After a parking turn left
Turn left onto the sidewalk and then continue for 220 m
The recreation ground area koupaliště Džbán is on your right
After the sports centre SK Aritma Praha turn right
Turn right away from a bridge and then continue for 650 m
The reservoir vodní nádrž Džbán is on your right
The target is in front of you

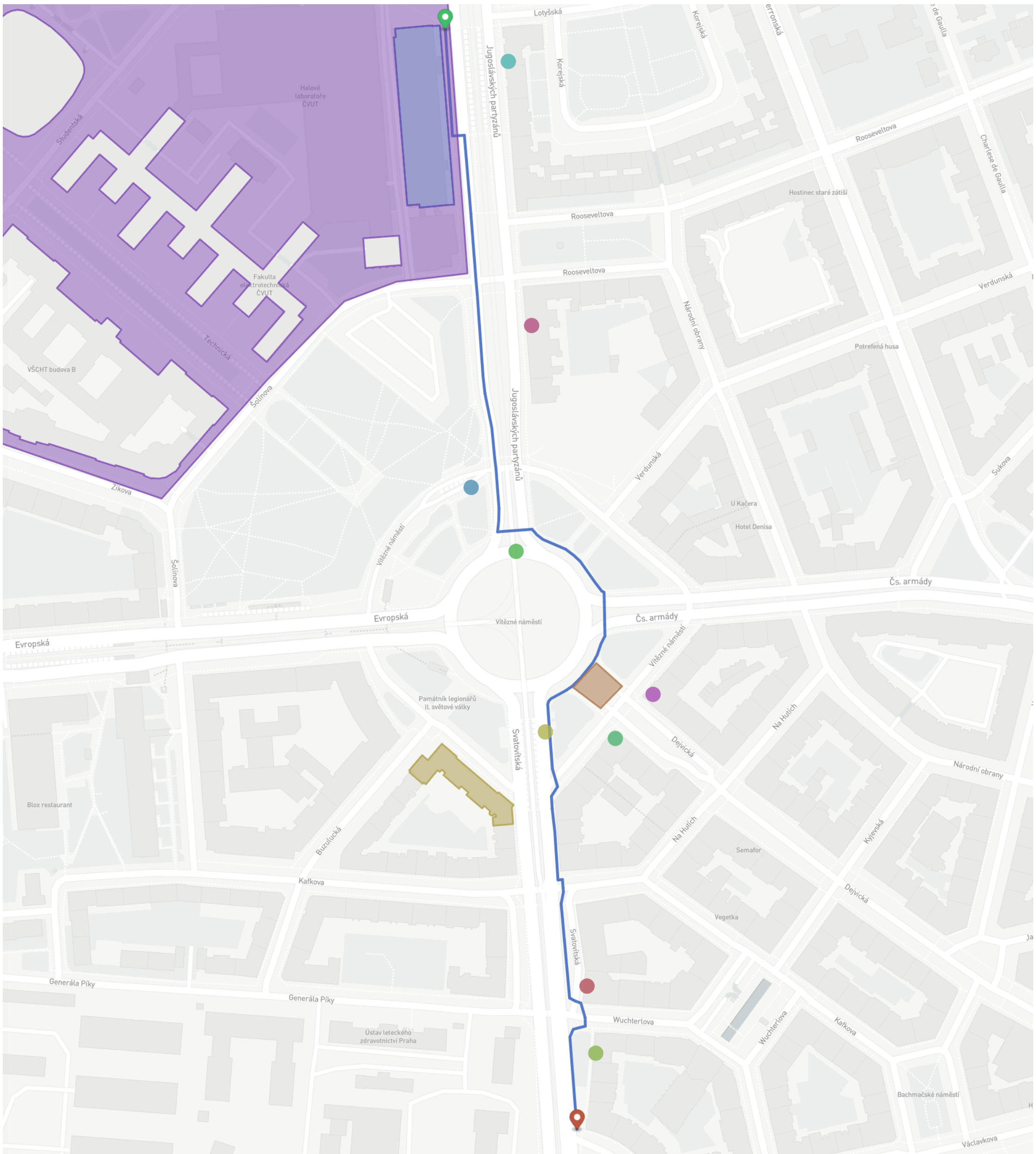


Figure A.3. Route 3

A.3 Route 3: Lotyšská – Kafkova

- Go towards the shop Nápoje Potraviny
- Continue straight away from the Český institut informatiky, robotiky a kybernetiky and then continue onto Jugoslávských partyzánů Street
- The university Kampus Dejvice is on your right
- The shop 5. avenue is on your left
- After the public transport platform Dejvická turn left
- Turn left over the crosswalk
- A tram crossing is on your right
- Turn slight right towards the cafe Café Šesták
- Continue straight by the fast food Bageterie Boulevard and then continue for 270 m
- The fast food KFC is on your left
- The public transport platform Vítězné náměstí is on your right
- The Dům armády is on your right
- At the restaurant Budvarka turn slight right
- Turn slight right towards the cafe Lavazza
- The target is in front of you

A.4 Route 4: FEL ČVUT – Studentský dům

- Go away from the university Fakulta elektrotechnická ČVUT
- Turn right by the ČVUT FEL, FS - budova T2 and then continue for 170 m
- The ČVUT FEL, FS - budova T2 is on your right
- Turn left towards the Národní technická knihovna and then continue onto Studentská Street
- Turn right towards the VŠCHT budova A
- Turn right away from the VŠCHT budova A
- Turn slight left away from the Národní technická knihovna
- Turn slight left towards the ČVUT FSv, FIT - budova A and then continue onto the service road
- The ČVUT FSv, FIT - budova A is on your left
- Turn right towards the university Studentský dům and then turn left
- Turn left away from the ČVUT FSv, FIT - budova A
- The target is in front of you

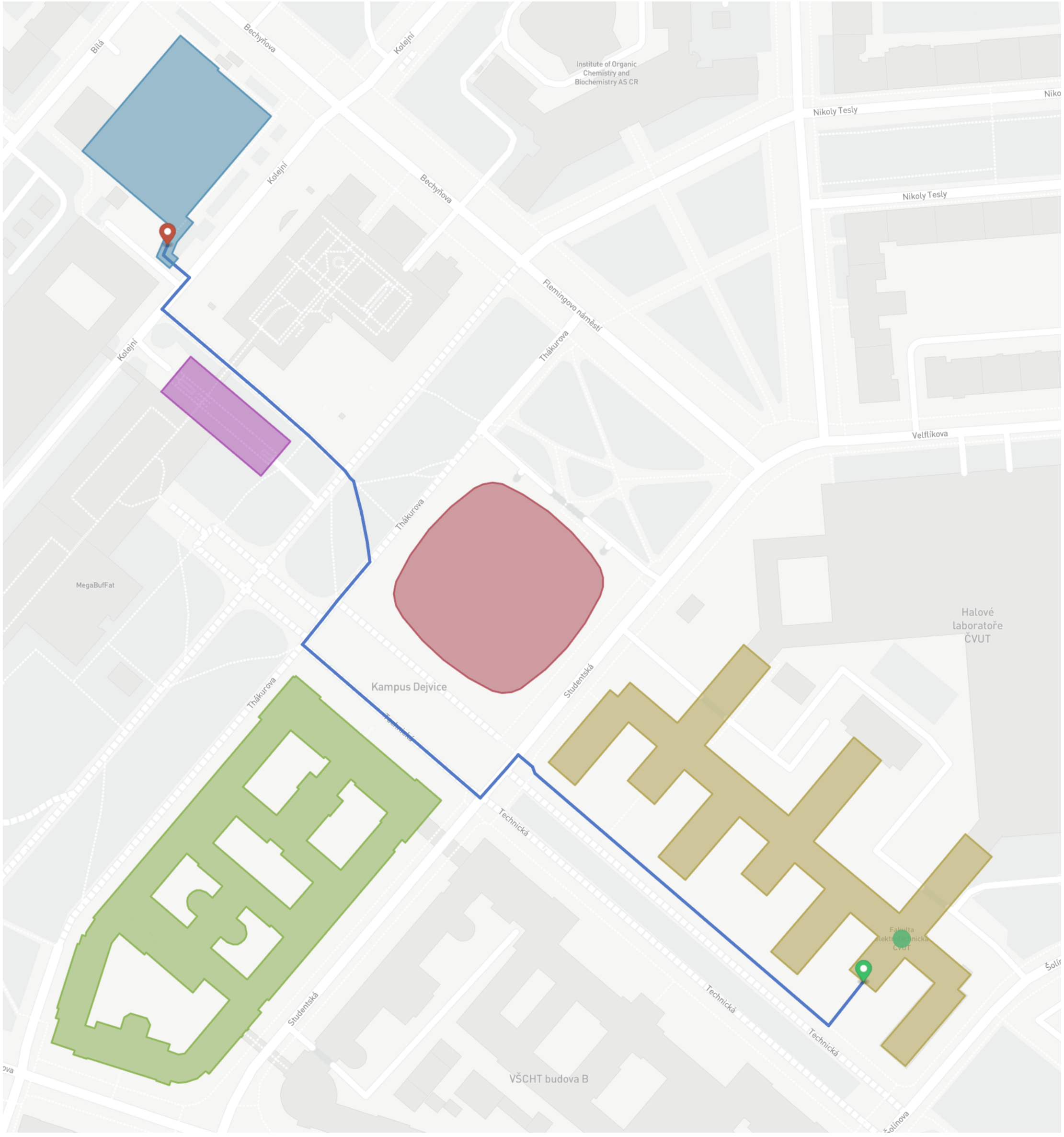


Figure A.4. Route 4

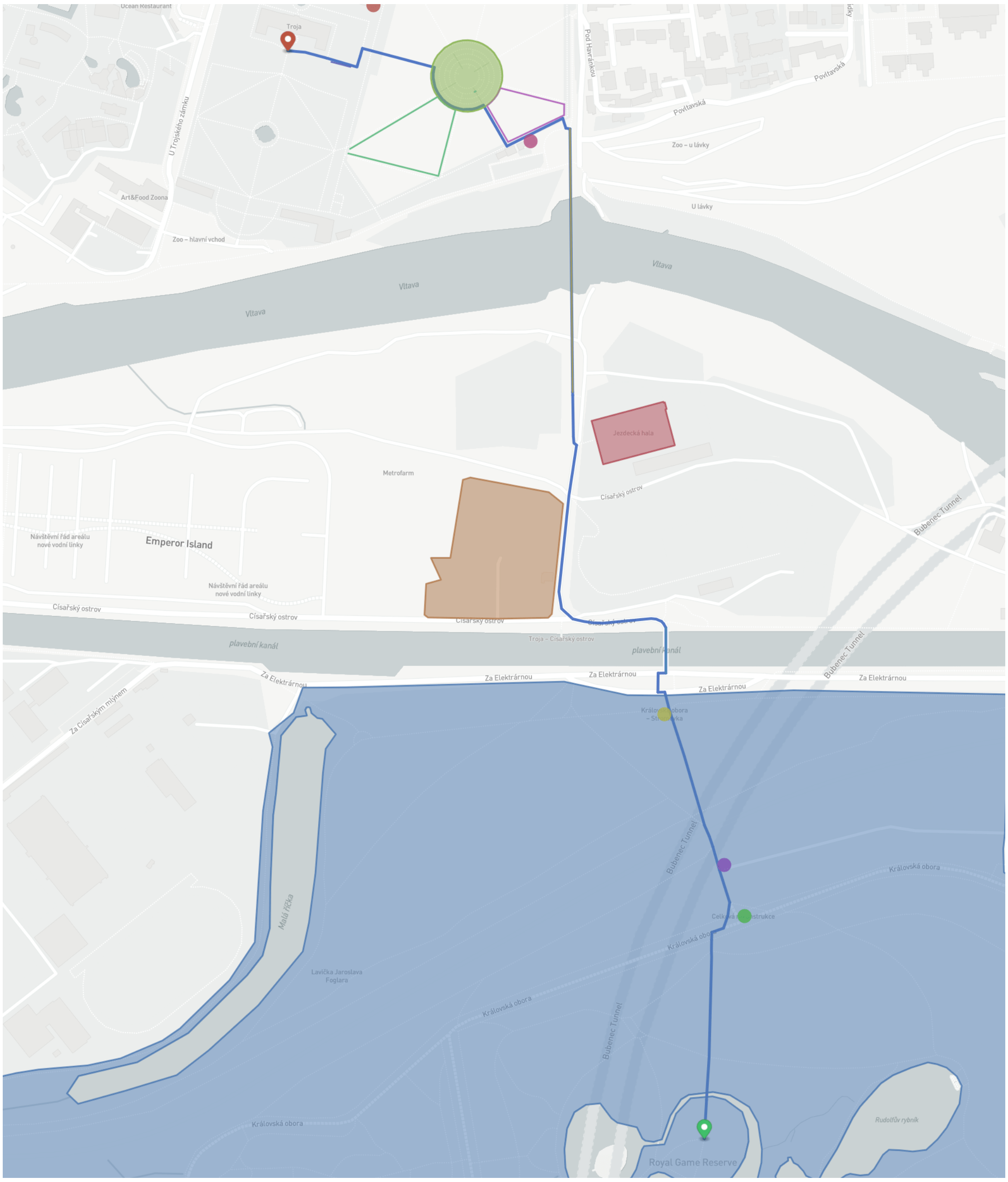


Figure A.5. Route 5

A.5 Route 5: Stromovka – Troja

Go north and then continue for 210 m

Continue over the bridge

At the T junction turn right

Turn right towards the information board Celková rekonstrukce

Turn slight left and then continue for 230 m

The information board Začlenění části výstaviště do areálu Královské obory STROMOVKA is on your right

At the map Královská obora - Stromovka continue straight

Continue straight away from the map Královská obora - Stromovka and then turn left

Turn left by the park Královská obora and then turn right

Turn right up the stairs and then continue onto Za Elektrárnou Street

Continue straight onto Za Elektrárnou Street and then continue over the bridge

Continue over a bridge

Turn slight right by a sports centre and then continue for 180 m

At the Jezdecká hala turn slight left

Turn slight left by the Jezdecká hala and then continue for 310 m

Continue over the Trojská lávka Bridge

After 200 m turn left

Continue over the bridge Trojská lávka

Turn left towards the fast food Občerstvení U lávky and then turn right

Turn right by the fast food Občerstvení U lávky and then turn left

Turn left by a hedge

Turn right towards a bridge

Turn left by a bridge

A hedge is on your left

Turn left by a castle garden

Turn left away from a power substation

A hedge is on your left

The target is in front of you

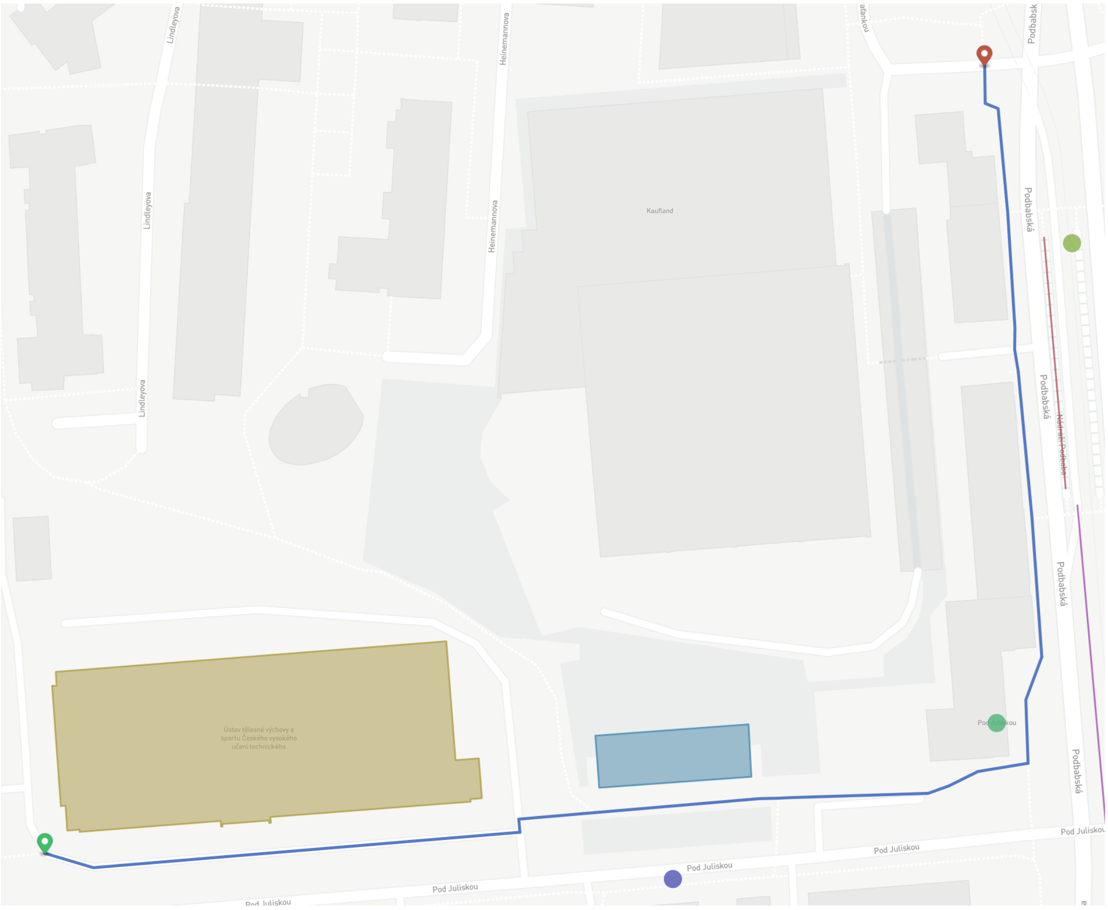


Figure A.6. Route 6

A.6 Route 6: Juliska – Podbaba

- Go towards the sports centre Ústav tělesné výchovy a sportu Českého vysokého učení technického and then continue for 130 m
- At the T junction continue straight
- Continue straight by the Stanice techniků dům hlavního města Prahy and then continue onto the footway
- A parking ticket machine is on your right
- Turn left towards the restaurant Pod Juliskou and then continue for 190 m
- The tram rails is on your right
- The target is after the tram stop Nádraží Podbaba
- The public transport platform Nádraží Podbaba is on your right
- The target is in front of you

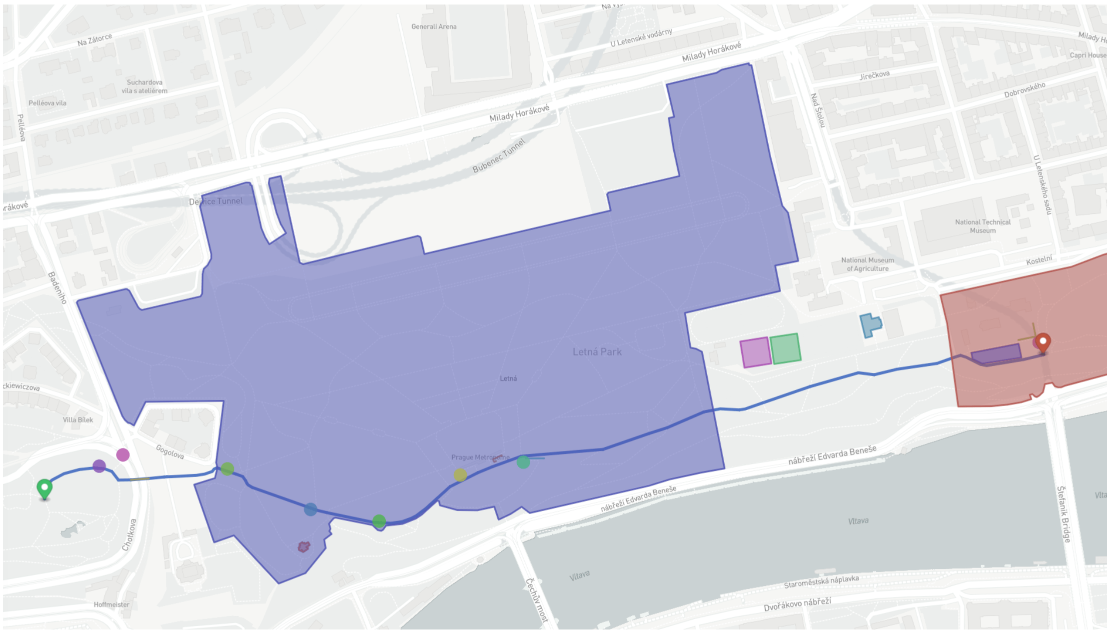


Figure A.7. Route 7

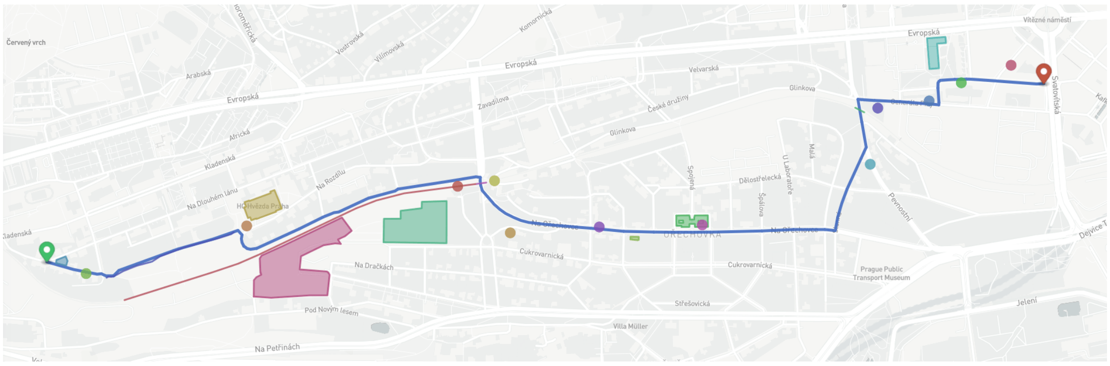


Figure A.8. Route 8

A.7 Route 7: Chotkovy sady – Letenské sady

Go north and then continue for 130 m
A bench is on your right
Turn slight left by a tram crossing and then continue for 170 m
Continue over a bridge
After the map Letenské sady continue straight
Continue straight away from the map Letenské sady and then continue for 730 m
The toilets is in front of you
The cafe Hanavský pavilon is on your right
A bench is in front of you
The pub Stalin Containall is on your left
A wall is on your left
A guidepost is on your right
Continue away from a retaining wall
After the park Letenské sady turn slight left
Turn slight left away from the park Letenské sady and then continue for 110 m
A pitch is on your left
Continue straight by a pitch and then continue for 330 m
A biergarten is on your left
The park Letenské sady is in front of you
At a hedge turn left
The biergarten Zahradní pivnice Letenské sady is on your left
Turn left towards a drinking fountain
The target is in front of you

A.8 Route 8: Veleslavín – Kafkova

Go towards a playground and then continue for 170 m
After a gate continue straight
Continue straight towards the stream Dejvický potok
Turn slight left onto the path and then continue for 390 m
The stream Dejvický potok is on your right
At the T junction turn right
Turn right by the stadium HC Hvězda Praha and then continue away from a bollard
Continue straight away from a bollard and then continue for 770 m
The shop Zahradnictví Chládek is on your right
A railway is on your right
The sports centre SK Střešovice is on your right
At a bridge turn right
A railway crossing is on your right
Turn right by a railway crossing
Turn slight left away from a bridge and then continue for 1,1 km
After 1,0 km turn left
The embassy Velvyslanectví Turecké republiky is on your right
Continue away from the public transport platform Pod Vyhlídkou
The Kafkova vila is on your right
After 500 m turn left
The Ořechovka is on your left
The cafe Bistro Ořechovka is on your left
At the four-way junction turn left
At the four-way junction turn left
Turn left onto Slunná Street and then continue for 260 m
After the public transport platform Dělostřelecká continue straight
Continue straight onto Pevnostní Street and then continue for 150 m
After a bridge turn right
Turn right by a bridge and then continue for 240 m
The cafe Klubovna is on your right
After a recycling container turn left
Turn left by a recycling container and then continue for 380 m
The The Blox is on your left
The college Střední odborné učeliště kadeřnické is on your right
A parking ticket machine is on your left
The target is in front of you

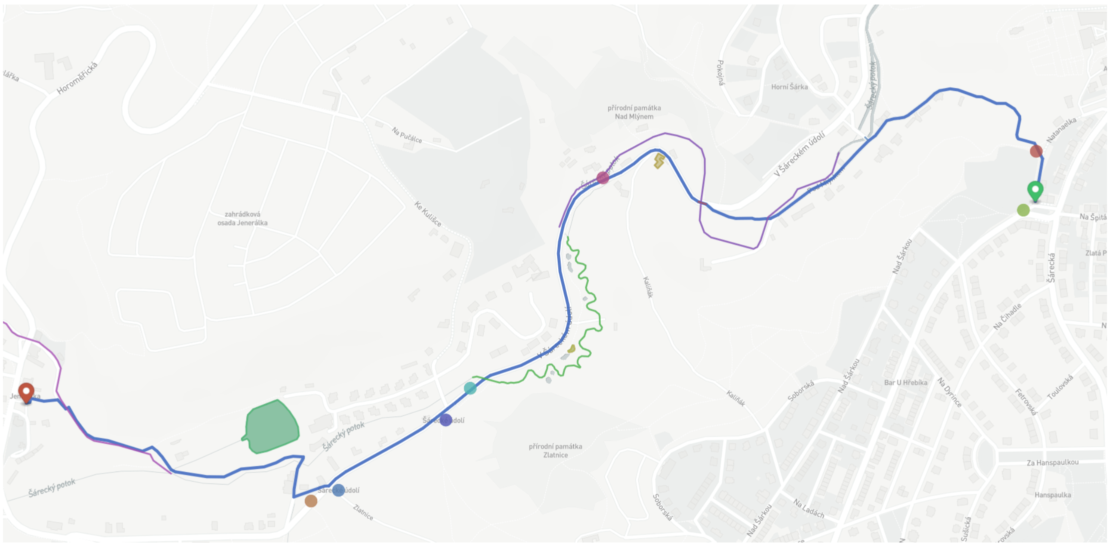


Figure A.9. Route 9

A.9 Route 9: Špitálka – Jenerálka

Go away from a recycling container
Turn slight left down the stairs and then continue for 610 m
The information board Tichá a Dolní Šárka is on your left
Continue downhill
The bridge Pod Mlýnkem is in front of you
At the Y junction turn slight right
Turn slight right by the stream Šárecký potok and then continue for 140 m
At the bridge V Šáreckém údolí continue straight
Continue straight over the V Šáreckém údolí Bridge and then continue for 220 m
At the Y junction turn slight right
The Vokorinka is in front of you
At the Y junction turn slight right
Turn slight right towards the public transport platform Kalinův mlýn and then continue for 810 m
A pond is on your left
The stream Šárecký potok is on your left
Continue over the V Šáreckém údolí Bridge
The public transport platform Kuliška is on your right
The guidepost Šárecké údolí is on your left
Before the public transport platform Korek turn slight right
Turn slight right away from the information board Šárecké údolí
Turn right and then continue for 360 m
Continue over the K Dubovému mlýnu Bridge
The pond Dubák is on your right
At the T junction turn left
Turn left by the stream Nebušický potok and then continue for 260 m
Continue over the bridge
The target is in front of you

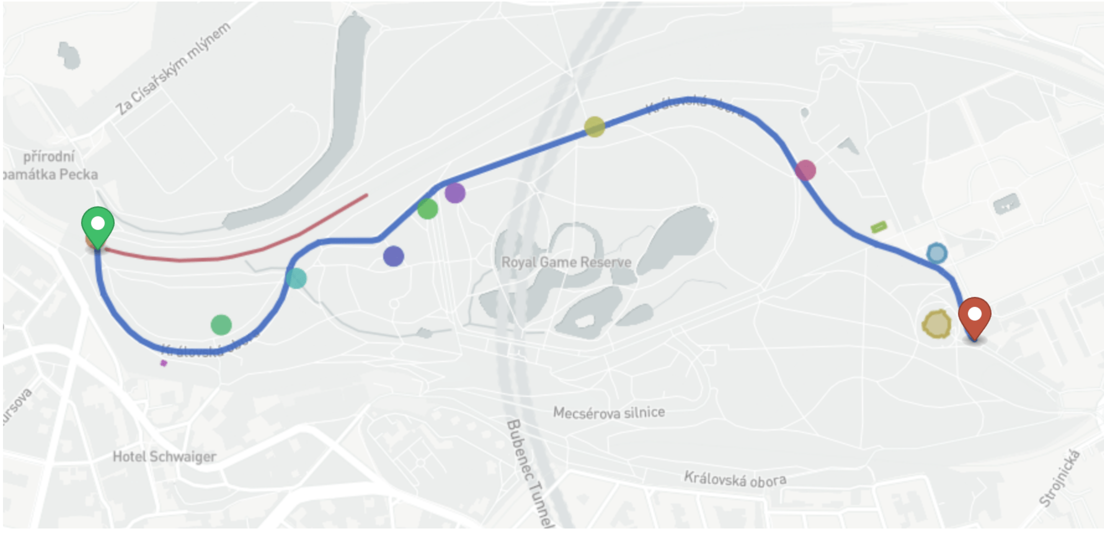


Figure A.10. Route 10

A.10 Route 10: Stromovka – Planetárium

Go away from the toilets and then continue for 2,1 km
The Studniční domek is on your right
A playground is on your left
A bench is on your right
A railway is on your left
A shelter is on your right
The toilets is on your right
The artwork Milenci is on your right
The information board Celková rekonstrukce is on your left
A gate is on your left
The restaurant Bistro NO. 2 is on your left
The museum Maroldovo panorama is on your left
The target is after the museum Planetárium
The target is in front of you

Appendix B
Source files

source.zip

- `java` - prototype application (Section 5)
  - `src` - Maven project source files
  - `javadoc` - documentation
  - `pom.xml` - Maven project file
- `osm2po` - osm2po tool for creating routable graph (Section 4.2)
- `osmosis` - osmosis tool for height data enrichment (Section 4.1.8)
- `Dockerfile` - document to build a Docker image (see 5.1)
- `import_db.sh` - BASH script to import OpenStreetMap data into PostgreSQL database (Section 4.2)
- `process_db.sh` - BASH script to enrich routing graph with context information (Section 4.3)
- `README` - text file with technical details
- `RUN.sh` - BASH script to automatically run the entire process from data import to launching the Web application (see 5.1)