

**ARPA: Accessibility-focused Route Planning
Assistant**

Igor Ershov, BSc

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

**Master of Science in Computer Science (Future Networked
Systems)**

Supervisor: Siobhán Clarke

August 2021

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Igor Ershov

August 31, 2021

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Igor Ershov

August 31, 2021

ARPA: Accessibility-focused Route Planning Assistant

Igor Ershov, Master of Science in Computer Science
University of Dublin, Trinity College, 2021

Supervisor: Siobhán Clarke

Moving around a city can be challenging for walking impaired people, with many preferring to adhere to already known roads or walk around only with a personal assistant. Different factors of the urban environment play into it, including types of kerb, sidewalk width, and presence of street furniture. Many of these factors affect different categories of people differently, and often the needs are contrasting. For example, while an older adult prefers having street furniture to rest at, wheelchair and blind users prefer encountering none.

Academic research has studied ways to offer routing solutions to navigate urban environments. Still, they often rely on private and hand-gathered data, or, if they use public datasets, they only work in small spaces that have been carefully mapped, not taking into account the unreliable or limited coverage of public data. On the other hand, publicly available solutions often offer limited customisation capabilities and do not consider the user's feedback into the route, only proposing the route the algorithm established as the best.

Building upon this established research, this dissertation aims to find out to what extent is it possible to create a routing algorithm that can provide routing to walking impaired persons, taking into account their specific needs, showing them

the accessibility information for each step of the route while also letting them have control of the final route, using publicly available datasets for accessibility information.

To achieve this, a customisable walking-focused routing algorithm that provides an individualised route that addresses their needs is presented. The accessibility information is gathered from the OpenStreetMap project. Users are prompted with questions about their needs, and the answers will then be used as weights for the A* algorithm used when creating the route. When making a route, all the accessibility information used will be shown to the user to account for missing or unreliable information. Based on this additional information, the user can veto specific road parts they do not feel comfortable taking and get offered alternative solutions until a satisfactory route is found.

Acknowledgement

I would like to thank my supervisor Professor Siobhán Clarke who provided valuable advice throughout the process of writing the dissertation. Many thanks also to the Crowd4Access team for giving me significant insight about OpenStreetMap.

I would also like to extend my gratitude to Philippa and Cathal for sticking with me throughout this Masters and working as a group to tackle numerous courses and group projects. In addition, I cannot begin to express my thanks to Chiara, Ivan, Valeria, Edoardo, and Febe, whose friendship carried me through this unprecedented year of isolation. Finally, I'm highly indebted to Martina for motivating me to work on my dissertation day after day in our virtual study sessions together.

Lastly, completing this dissertation would not have been possible without the support and nurturing of my family.

IGOR ERSHOV

*University of Dublin, Trinity College
August 2021*

Contents

Abstract	iii
Acknowledgement	v
Chapter 1 Introduction	1
Chapter 2 Background	4
2.1 Accessible cities	4
2.1.1 Pedestrian focused cities	4
2.1.2 Building for accessibility	6
2.1.3 Mobility and Walking Impairments	7
2.1.4 Requirements for people with walking impairments	8
2.1.5 Further discussion on other mobility impairments	13
2.1.6 Routing as a way to help people	14
2.2 Crowdsourced mapping	15
2.2.1 Crowdsourced Geographic Information	15
2.2.2 OpenStreetMap	16
2.2.3 Reliability and accuracy concerns	20
Chapter 3 State of the Art	22
3.1 Overview	23
3.2 A review of " <i>Collaborative personalized multi-criteria wayfinding for wheelchair users in outdoors</i> "	27
3.3 A review of " <i>On Combining Crowdsourcing, Sensing and OpenData for an Accessible Smart City</i> "	29

3.4	Commercial applications	31
3.5	Routing based on privately gathered data	33
3.6	Routing based on VGI	35
3.7	Other approaches	37
Chapter 4 Design and Implementation		39
4.1	Project overview	39
4.1.1	Functional architecture	39
4.1.2	Application flow	40
4.1.3	Technology choices	42
4.2	Gathering the input	43
4.2.1	User weights	43
4.2.2	Geocoder	45
4.2.3	Map data	46
4.3	Creating the graph	47
4.3.1	Parsing the data	47
4.3.2	Creating the graph	49
4.3.3	Gathering accessibility information	49
4.4	Routing Algorithm	54
4.4.1	Adding start and end points	54
4.4.2	Routing algorithm	54
4.4.3	Compacting the result	57
Chapter 5 Evaluation		58
5.1	Comparison against research question	59
5.2	Comparison of the general quality of generated routes	61
5.2.1	Explanation of comparison metrics	62
5.2.2	Comparison environments	65
5.2.3	Adapting applications for evaluation	66
5.2.4	Results and discussion	70
5.3	Discussion	72
Chapter 6 Conclusion		73

Bibliography	76
Appendices	87
Appendix A Screenshots of the application	88
Appendix B Constants and values used in the routing function	91

List of Tables

2.1	<i>A table summarising the accessibility requirements of wheelchair users and the respective recommended values</i>	11
2.2	<i>A table summarising the accessibility requirements of older people and the respective recommended values</i>	12
3.1	<i>A table comparing the different features offered by the approaches covered in the state of art</i>	24
3.2	<i>A table comparing the different accessibility requirements tackled by the approaches covered in the state of art</i>	26
4.1	<i>A table outlining the questions used to gather the user's preferences towards the different accessibility features</i>	44
4.2	<i>A table outlining the decimal equivalent of the possible responses to the questions</i>	45
4.3	<i>A table outlining the accessibility features gathered from Nodes, with the queried tags and the values saved</i>	51
4.4	<i>A table outlining the accessibility features gathered from Ways, with the queried tags and the values saved</i>	53
5.1	<i>Parameters used in the API call to the Wheelchair Open Route Service API for the wheelchair generalised evaluation</i>	66
5.2	<i>Weights used for ARPA for the wheelchair generalised evaluation</i>	67
B.1	<i>A table outlining the constants used in the routing function</i>	91
B.2	<i>A table outlining the decimal value equivalent of the surface smoothness</i>	92

B.3	<i>A table outlining the decimal value equivalent of the track types . . .</i>	92
B.4	<i>A table outlining the decimal value equivalent of the kerbs</i>	92
B.5	<i>A table outlining the decimal value equivalent of the crossing types .</i>	92
B.6	<i>A table outlining the decimal value equivalent of the surface materials</i>	93

List of Figures

2.1	<i>Example of a Node represented as an XML element</i>	18
2.2	<i>Example of a Way represented as an XML element</i>	19
2.3	<i>Example of a Relation represented as an XML element</i>	19
4.1	<i>Sequence Diagram outlining the flow of the application when a user queries for directions</i>	41
4.2	<i>Equation to determine the accessibility cost of the surface type of a Way</i>	56
4.3	<i>Equation to determine the accessibility cost of the kerb height of a Way</i>	57
4.4	<i>Equation to determine the accessibility cost of the crossing type of a Way</i>	57
5.1	<i>Equation to determine the reliability value of a piece of accessibility information</i>	63
5.2	<i>Equation to determine the reliability score of a single segment . . .</i>	63
5.3	<i>Equation to determine the reliability score of a route</i>	63
5.4	<i>Equation to determine the accessibility value of a piece of accessibility information</i>	64
5.5	<i>Equation to determine the accessibility score of a single segment . .</i>	64
5.6	<i>Equation to determine the accessibility score of a route</i>	64
5.7	<i>Equation to determine the total score of a route</i>	65
5.8	<i>Graph showing the results for the generalised evaluation without requirements</i>	70
5.9	<i>Graph showing the results for the generalised evaluation with the requirements set for a wheelchair user</i>	71

A.1	<i>Screenshot of the main page of the application</i>	89
A.2	<i>Screenshot of the weights page of the application</i>	90

Chapter 1

Introduction

Modern cities are built by people without disabilities for other people without disabilities. Consequently, the experience of traversing a city changes drastically according to the mobility capabilities of a person. Situations where impassable roadblocks prevent mobility-impaired pedestrians from efficiently traversing pedestrian roads are commonplace. For example, a typical road feature like a kerb can present a difficult obstacle for wheelchair users without a proper ramp to allow access to it. Meanwhile, a street crossing without appropriate audio signals can force a blind pedestrian to take another road as they cannot safely cross. Without proper planning and design, cities themselves can become what restricts people with disabilities from achieving their full potential.

Mobility-impaired people are all the people whose functional limitations affect their ability to move independently. This categorisation covers a broad set of physical, cognitive and sensory impairments. Because of the diversity of impairments, the requirements that could make a road accessible for some people might be an obstacle for others. For example, while older adults prefer encountering street furniture in their path to rest at, those act as obstacles for a wheelchair user going on the same road.

While the solutions for the problems in urban design need to come from legislation and standardisation of accessibility-focused design, technology can help limit

the impact of thoughtless urban planning for mobility-impaired people. The approach explored in this dissertation uses technology to provide mobility-impaired people with directions for their daily trips in urban environments that consider their specific walking abilities to avoid inadequate roads. Such an approach is not unusual, however, technical limitations make mobility-impaired prefer using map routers as planning tools to determine the conditions of roads before deciding on which path to take rather than blindly trusting the directions provided.

Ample research has been put into urban routing for people with reduced mobility. However, Academic research often relies on private and hand-gathered data or is only available for small, carefully mapped areas when using publicly available datasets, and by doing so does not take into consideration the unreliability and limited coverage of public data. On the other hand, while they usually have more coverage, commercial applications offer fewer customisation options and hide the accessibility features of the road from the user, limiting their usefulness.

Building upon the current state of the art, this dissertation aims to establish to what extent it is possible to create a routing application with the following requirements:

1. Generate directions suitable for pedestrians in a short amount of time.
2. Take into account the user's specific needs.
3. Provide users with accessibility information for every step of the route.
4. Allow users to have control of the final generated route.
5. Use publicly available datasets for road and accessibility information.

To answer this research question, this dissertation presents ARPA, an Accessibility-focused Route Planning Assistant. ARPA is a customisable walking-focused routing algorithm that provides an individualised route that addresses a user's specific requirements. It uses OpenStreetMap, a crowdsourced geographic database, as the source of the accessibility information of road networks. The routes are generated

with an A* algorithm that compares the user's requirements with the road's accessibility features to generate appropriate directions. Furthermore, the users are provided with all the accessibility information of the generated directions, from which they can veto any specific road they do not feel comfortable using and get offered alternative solutions until a satisfactory route is found.

Structure of the dissertation

The remainder of the dissertation is structured as follows.

The second chapter provides context for the project and is split into two parts. The first part introduces the motivation behind the dissertation and outlines the requirement of mobility-impaired people. The second part introduces the OpenStreetMap project, explains the data structures referenced throughout the dissertation and discusses the benefits and drawbacks of using a crowdsourced geographic database.

The third chapter outlines the state of the art of pedestrian-focused routing applications, examining to what extent the various academic and commercial approaches tackle the requirement outlined in the research question.

The fourth chapter details how ARPA operates by explaining the various steps of the routing process. It starts with an overview of the application, then explores the gathering of the input, the creation of the road network graph and the generation of the most suitable route.

Finally, in the fifth chapter ARPA is evaluated against two commercial applications, Google Maps and OpenRouteService. The three applications are compared by the extent they tackle the research question and the overall quality of the generated routes.

Chapter 2

Background

2.1 Accessible cities

2.1.1 Pedestrian focused cities

With the popularisation of cars, cities started to grow more towards the concept of urban sprawl (1). Instead of neighbourhoods that provided everything that a person would need, companies and shops began opening where the land was cheapest, assuming that their employees and customers would have a car (2). With time *automotive cities*, cities that facilitate and encourage the movement of people via private transportation, became the standard way to build cities. Roads stopped being communal places where every person had equal right to be on them as they gave way to the new monopoly of cars (3).

This way of building cities is not sustainable as it has numerous health, social and environmental concerns. Urban sprawls are correlated to increased health risks due to air pollution that leads to breathing problems, and they are linked with obesity concerns on account of a reduced rate of outdoor activities by their citizens ((4),(5)). They also reduce social interactions as citizens will travel further to meet their basic needs, therefore removing the concept of neighbourhoods and fostering exclusion and inequality between the various districts ((6),(7)). Furthermore, car-centric urbanisation plays a significant role in global warming, with 75% of the population in developed countries living in urbanised areas and trans-

portation making up 29% of the global CO2 emissions ((8),(9)). For these reasons, there has been an increased push to take back the streets and make pedestrians the focus in recent times.

The most effective ways for urban planning to shift towards more livable cities have been a combination of different types of initiatives (10). The first is to offer more sustainable modes of commuting to citizens with solid public transport infrastructures. The second is the creation and incentivisation of the use of novel types of transportation inside of cities, such as e-scooters and public bikes scattered across city centres, combined with the transformation of said city centres into car-free zones, limiting some of the most popular parts of cities to pedestrians and cyclists (11). The third is to remove their transport needs at all, which is being tackled by the popularisation of city planning ideas such as the 15-minute neighbourhood, where all the basic facilities needed by a citizen should be within a 15-minute reach from residential districts (12). This global push has led to a hopeful pattern in industrialised countries, where there has been a steady decline of young people taking driving licences after the country reached their *peak car* moment (13).

While urban planning is one of the most important aspects of making a city more livable, other aspects are needed to make a city more alive. It is usually not enough for a road to be available and well designed to make it a great street. In his research, Alfonzo M. (14) argues about the existence of a *Hierarchy of Walking Needs* involved in the decision-making process that leads up to the choice of walking to a destination, or as a recreational activity. This hierarchy features at its base the Feasibility and Accessibility of the route, which are tackled by the urban planning techniques outlined previously. Next, it introduces the requirements of a feeling of Safety and Comfort, which are associated with noise levels and the perception of upkeep and maintenance of the road and the surroundings. Finally, the Pleasurability of a road is analysed, explaining the importance of a varied cityscape and features such as parks and large public spaces. These considerations are often abstracted in the concept of *Walkability* (15) to evaluate the quality of living associated with a road and as a tool for urban planners to compare different

neighbourhoods and identify faults in the city. Because of its abstract nature, there is no objective way to measure Walkability, with much discussion about the best approaches towards measuring it (16). One common finding across all the indices is that neighbourhoods with high Walkability values are often associated with increased wealth (17), health (18), and sustainability (19), highlighting the importance of environments focused on the pedestrian's quality of life.

2.1.2 Building for accessibility

The experience of traversing a city can change drastically according to the mobility capabilities of a person. People without disabilities typically build cities with other people without disabilities people in mind, leading to situations where insurmountable roadblocks prevent people with disabilities from efficiently traversing pedestrian roads. A typical road feature like a kerb can be a tricky obstacle for wheelchair users without a proper ramp to allow access to it, or a street crossing without appropriate audio signals can force a blind pedestrian to take another road as they can not safely cross. Cities themselves can become crippling to people with disabilities who are restricted from achieving their full potential due to poor planning and design (20).

Around 15% of the world's population, which makes up 785 million persons, has been diagnosed with some form of disability, with 2.2% having significant difficulties in their day to day life because of it. The percentage is only expected to rise with the ageing of the population and the spread of chronic diseases such as diabetes and mental health issues. Different types of disabilities affect people in multiple ways, as no impairment has the same impact. For example, on average, children with physical impairments fare better in academic contexts than children with sensory or mental impairments. Furthermore, people with cognitive or intellectual impairments are among the more marginalised groups in the labour market (21).

While there has been an effort to standardise and make cities more accessible, these are long term goals that will require action from each city's urban plan-

ners. Examples of efforts underway are the focus on accessibility in the UN 2030 Agenda for Sustainable Development (22) and an increase in awareness initiatives (23). Today, studies have shown that most trips around big urban centres like London have increases of up to 50% in trip length if in the presence of people with mobility impairments as they will need to be rerouted to avoid certain inaccessible areas or blocking features in the environment ((24),(25)). Smaller and more focused studies have also shown the problems in enclosed spaces such as train stations (26) and shopping centres (27), all with similar results.

Many scholars argue that current urban areas fail to deliver essential commodities to people with disabilities and that this is an infringement on the civil rights of a significant part of the population (28). Often environments are described by being characterised by *design apartheid* (29), where the way environments are built mirror the actual *disablist* values of the societies that inhabit them (30). Even though planning legislations are present, they are often poorly designed or outright ignored and not enforced (31).

2.1.3 Mobility and Walking Impairments

As described by the World Health Organization, disabilities are a complex phenomenon that comes from the relationship between a person's physical capabilities and their community (21). The scope of this dissertation focuses on an application to facilitate the routing of walking impaired people. Walking Impairments are a subset of Mobility Impairments, which are defined by Thorsten Völkel et al. as :

"Mobility impairments include all functional limitations which affect the ability of a person to reach a remote destination independently. A physical, cognitive, or sensory impairment or a combination of them may lead to mobility impairment."(32)

This definition covers a broad set of people with either permanent or temporary impairments that limit their independent movement in an environment. Some of the groups included in the definition are visually impaired and blind people, people

with cognitive impairments, wheelchair users, older people, and deaf people, each individual often having a specific set of requirements. For example, older pedestrians would prefer to avoid steps and favour routes with plenty of street furniture to rest. Blind or pedestrians with low vision would select courses with tactile paving and intersections with acoustic cues; furthermore, these routes are usually learned with the assistance of orientation instructors and rarely change. Pedestrians with cognitive impairments typically need to be assisted as issues such as short-term memory loss could impact their sense of orientation. Finally, wheelchair users need paths with even surfaces, kerbs provided with ramps and wide footpaths (33).

A consideration that needs to be made is that because every individual has different preferences and requirements, these might also clash with those of other people. For example, while older people would prefer having street furniture in their path, those would provide obstacles for wheelchair users. A similar common conflict is between blind pedestrians and wheelchair users concerning footpath height; while blind pedestrians would prefer those to be as high as possible to distinguish the sidewalks from the road quickly, wheelchair users would like them to be closer to the ground as it would allow easier access for them (34).

As mentioned before, the focus of this dissertation is to assist in the creation of routes for walking impaired people, which are a subsection of mobility-impaired people whose physical characteristics restrict their freedom of movement in urban environments. This distinction is made because some mobility impairments require additional considerations not covered by the research question. These considerations are further explained in section 2.1.5.

2.1.4 Requirements for people with walking impairments

An extensive literature review has been carried out to understand the various requirements that different walking impaired people have in their route choices. Essential sources have been the DIN18024-1 (35), a German specification that features guidelines for creating accessible spaces for various types of impairments, and Americans with Disabilities Act (36), the American equivalent, which features

similar values. Furthermore, other studies have conducted comprehensive surveys on the mobility impaired population, leading to numerous insights. A set of requirements for walking impaired people derived from the literature review will now be outlined.

Wheelchair users

Wheelchair users are some of the most vulnerable people when it comes to urban design; therefore, numerous surveys and studies have been carried out to understand their needs and requirements. The following are the requirements of road features for wheelchair users, as established by the frequency of mentions and the results of the surveys and studies from the literature review.

- In the presence of stairs, an adjacent ramp should always be present. This ramp should feature handrails and should include level resting places on long ramps ((35),(36)).
- In the presence of a raised sidewalk, the kerb should be as low as possible, with a recommended maximum height of 3cm ((35),(36),(37)).
- Access to the kerb should be provided with access ramps ((35),(36),(34)).
- Road segments and ramps should not present steep gradients, with the maximum recommended slope inclination being 3-6% ((35),(36)).
- The road surfaces should be made of a material that should not hinder wheelchairs, with the best materials being Concrete and Paving, and the worst being Gravel and Cobbled surfaces ((35),(37),(38),(39)).
- The edges of footpaths should not present deep gutters where wheels could get stuck ((37),(20)).
- The width of sidewalks should be at least 91.5 cm ((35),(36)).
- Sidewalks should reduce the use of adverse cambers as much as possible ((37),(20)).

- Street furniture should not be placed in a way that would not allow at least 91cm on either side to allow passage ((37),(20),(40)).

Moreover, some common hindrances that weigh into the choice of the route for wheelchair users are general traffic levels of cars, bicycles and pedestrians, the lack of overhead roofs, the presence of grating, and lack of street lighting ((37),(20),(40)). Furthermore, there is a strong preference towards roads close to accessible parking spaces and accessible facilities such as toilets (37).

A further consideration to be made is that these guidelines, recommendations, and hindrances are a generalisation of the overall average population of wheelchair users. Individual characteristics and type of wheelchair (manually assisted, self-propelled or motorised (37)) will increase or reduce the impact of the various points listed above.

Older Adults

No guidelines as strict as the ones for wheelchair users are present for older adults. Many of the features that help the former greatly enhance the latter's experience, making them very compatible. However, the mobility of older adults is generally better, causing some of the requirements to become preferences rather than necessities. The main point of conflict lies with street furniture, where wheelchair users would prefer to minimise their presence, older adults generally find them helpful to use as spots to rest (26). Surveys and studies carried on the maturer population have instead found general preferences that incentivise them to walk. Qualities such as noisiness, safety and presence of greenery in the path significantly increase the chances (37). Furthermore, they prefer going on routes with a low amount of traffic as they consider it safer, and they like the presence of street crossings with traffic lights and audio signals rather than uncontrolled ones (34).

While the focus of this section has been on wheelchair users and older adults, the concept of a walking impaired person includes a vast amount of people that are limited in their independence of movement in any physical way. Each indi-

Requirement	Description	Recommended value
Sidewalk	Presence of a dedicated footpath	Pedestrian-only
(Sidewalk) Kerb Height	Height of the kerb in regards to the road level	<3cm
(Sidewalk) Inclines	Gradient incline of sidewalk/road segments and ramps	<3-6%
(Sidewalk) Camber	Horizontal incline of the sidewalk/road	None
(Sidewalk) Width	Width of the sidewalk/road	>91.5cm
Pavement Type	Sidewalk/road surface material	Concrete
Steps	Presence of steps without an access ramp	None
Handrails	Availability of handrails at a ramp	At every ramp
Car traffic	Amount of car traffic on the adjacent road	Low
Pedestrian/Bicycle traffic	Amount of pedestrian and bicycle traffic on the sidewalk/road	Low
Street furniture	Presence of street furniture such as signs, benches, gutters and gratings	None
Street lighting	Presence of lighting on the sidewalk/road	Well lit
Amenities	Presence of amenities such as accessible parking and toilets	Frequent

Table 2.1: *A table summarising the accessibility requirements of wheelchair users and the respective recommended values*

Requirement	Description	Recommended value
Sidewalk	Presence of a dedicated footpath	Pedestrian-only
(Sidewalk) Inclines	Gradient incline of sidewalk/road segments and ramps	<3-6%
(Sidewalk) Width	Width of the sidewalk/road	Comfortably wide
Noise	Level of noise along the sidewalk/road	Low
Steps	Presence of steps without an access ramp	None
Handrails	Availability of handrails at a ramp	At every ramp
Street crossings	Type of street crossing	Supervised (Traffic lights)
Car traffic	Amount of car traffic on the adjacent road	Low
Amenities	Presence of amenities such as accessible parking and toilets	Frequent
Street furniture	Presence of street furniture such as benches	Frequent
Street lighting	Presence of lighting on the sidewalk/road	Well lit

Table 2.2: *A table summarising the accessibility requirements of older people and the respective recommended values*

vidual will have specific characteristics that are not generalisable but nonetheless important. This includes, but is not limited to, people using crutches, people with strollers and parents with children.

2.1.5 Further discussion on other mobility impairments

Numerous other types of impairments exist besides walking impaired people. Different kinds of impairments can, for example, be intellectual or sensory. However, those are not covered in this dissertation as their navigation in urban environments requires additional technological and accessibility considerations that fall outside the scope of the research question. Nonetheless, ample research has been done in the field, some of which is reported below.

One of the most common and disruptive sensory impairments are those of the visual kind, which cover vision loss and numerous eye disorders. The definition given by the International Council of Ophthalmology encompasses any visual and colour loss, along with complete blindness (41). Visually impaired people rely on route knowledge to navigate urban environments, depending on regular routes often chosen with an assistant to minimise complexity and obstructions (42). However, even familiar paths can sometimes present dynamic and unexpected obstacles such as construction work, overgrown trees and large puddles of water (43). Navigation aids such as dedicated guide dogs and canes are used to adapt to these dynamic changes with varying degrees of effectiveness (44). Another popular navigation aid are maps tailored to visually impaired people, with most of the development divided into four main categories (45): Virtual acoustic maps that represent map features as an audio output (46), virtual tactile map in which haptic feedback is used to explore maps (47), tactile-audio maps which are a combination of the previous two (48), and Braille tactile maps which use Braille to display map information through raised pins (49). Furthermore, there is a growing trend in the research of Augmented Reality technology as a means to guide visually impaired people by escorting them through the spatialised audio rendering of objects detected in their surroundings (43).

Another important category of people with mobility impairments is that of people with cognitive disabilities. Due to their reduced mental capability, they present limitations in their navigation of urban environments. Therefore, they often rely on set routes created by caregivers who also escort them for a period of time to teach them the directions and reference points on the path (50). Furthermore, they require mechanisms to quickly obtain assistance at their location due to the risks of losing their way combined with the difficulty they face in explaining their position (51). To assist in the navigation of people with cognitive impairments, novel approaches to maps and routing need to be considered. For example, AssisT-OUT (52) uses street-level photographs and descriptive text to guide users along their path, combined with quick access to caregivers contacts. Another approach is to use Augmented Reality technology to overlay directions over the real world view of the users to guide them to their destination (53).

2.1.6 Routing as a way to help people

Using technology to better the lives of people has always been a crucial topic of research. This is further highlighted when discussing people with disabilities; many researchers argue that as we have the technology to achieve extraordinary feats such as putting a man on the moon, we already have the tools to aid people in need (54). Nevertheless, as previously seen in the background chapter, many of the limitations of mobility-impaired people come from the thoughtless design of the environment that surrounds them, and as such, the solutions often need to first come from changing society rather than introducing new technology. However, societal change usually requires ample time and commitment, and technology can help lessen the impairments' effects while other people push for the broader changes.

For this, many different approaches have been introduced with time. Some breakthroughs are screen readers and voice-controlled software and devices for people with visual impairments, electronic wheelchairs for walking impaired people and assistive technology for people with cognitive disabilities ((55),(56)). This dissertation focuses on helping people with walking impairments move around

urban environments, possibly lessening the effects of accessibility barriers by providing a routing tool that they are able to reference before and during trips in their daily life.

2.2 Crowdsourced mapping

2.2.1 Crowdsourced Geographic Information

Crowdsourcing is a way to obtain information or help with a task from a large number of contributors. The term was first coined in 2004 as a portmanteau of the words *crowd* and *outsourcing* (57), and since then has seen rapid growth as a practice due to the widespread availability of the Internet. An examination of the literature by González-Ladrón-de-Guevara and Estellés-Arolas produced the following, more comprehensive, definition of crowdsourcing:

“Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of varying complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills. At the same time, the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken” (58).

As the dissertation focuses on accessibility in urban environments, there is a need for data that accurately represents the urban landscape. Because private companies show no interest in gathering the data, mapping the cities’ accessibility falls on the shoulders of the citizens themselves. This volunteer gathering of mapping information is labelled as *Volunteered Geographic Information* (VGI). It is formally defined as *the harnessing of tools to create, assemble, and dissem-*

inate geographic data provided voluntarily by individuals (59). VGIs, therefore, share many similarities with crowdsourcing, and it could be argued that they are a subsection of the more comprehensive crowdsourcing definition. The commonly agreed-upon distinction between the two terms relies on the accuracy of the information; Crowdsourced information implies a consensus between several people and, therefore, will have a higher likelihood of being accurate when compared to VGIs, which can be produced by single individuals (60). Section 2.2.3 covers the reliability and accuracy concerns of VGIs in more detail.

One of the most significant advantages of VGIs is that, as local volunteers collect them, they can contain local knowledge and specific information that can not as be gathered as readily through more traditional means (59). This has been a crucial advantage in the widespread use of VGIs for many applications. The most prolific is disaster management, where citizens can quickly provide information on local conditions in a disaster or crisis, with the most widespread uses being floods and fires (61). VGIs have also been extensively used to carry out transportation studies (62), assess environmental concerns (63) and, most close to this dissertation, gather accessibility information.

The two most notable projects in this last category have been WheelMap and OpenStreetMap. OpenStreetMap is used as the primary data source for this dissertation and, as such, is covered in more detail in the next section. Wheelmap, instead, is a map of wheelchair accessible places developed by Sozialhelden (64). It uses a semaphore system to signify if a point of interest is wheelchair accessible or not, making it easy for volunteers to assess points of interest and for users to determine accessible locations.

2.2.2 OpenStreetMap

Geographical data is, in most cases, private and under strict restrictions for its uses. Even if public entities funded by taxpaying citizens gather the data, it is often not available to the citizens without additional licensing fees payments for licences, or if it is available, it is often crude data. Moreover, licences often contain

extreme restrictions that limit their uses, restricting the applications that can be developed with them even if the due fees are paid. Furthermore, the price tag does not guarantee an assurance of quality. Copyrighted maps can quickly become outdated and contain misleading information known as copyright easter eggs, used to determine if plagiarism has happened between different map distributors.

These limitations were what inspired the creation of the OpenStreetMap (OSM) project (65). The website defines the project as "*[...] a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license*" (66). The project aims to map the globe and distribute the data under permissive copyright schemes.

Steve Coast founded the project in 2004 as a response to government agencies, such as the British Ordnance Survey, creating massive map datasets from public funding but failing to distribute them. The project started at University College London, which still maintains and hosts the main infrastructure to date. In 2006 the OpenStreetMap Foundation was created to promote the growth and development of the project and to provide free geospatial data for anyone to use and distribute (67).

To date, the project is increasingly becoming more popular. At the time of writing, it is approaching 8 million contributors and 9 billion GPS data points. A phenomenon becoming more popular in recent years is the rise of corporate contributors, people hired by big corporations like Amazon, Microsoft and Apple to contribute to the project as they become more reliant on it for their day to day operations. These contributions started after many big corporations involved with mapping stopped supporting their private initiatives, instead joining the OpenStreetMap project. This increase in corporate interest has sparked controversy and discussion in the community as there is an increasing worry of a corporate takeover of the project (68).

Database elements

The main file format of the OSM project is XML, chosen because it is human-readable, machine-independent and has a good compression ratio (69). Every data point in OSM is represented as an *Element*, which are the basic components of the OSM representation of the physical world. There currently exist three different types of Elements: *Nodes*, *Ways*, and *Relations*.

Nodes represent any single point in space and are defined by their latitude, longitude and ID. The most common use of a Node is to define the shape of a Way and can be part of many Ways at once. When a Node is shared across multiple Ways, it represents an intersection between them (for example, in a road junction). Nodes are also used on their own to describe features in the environment, and when used in this manner, they have at least one tag to define their purpose (70).

```
<node id="25496583" lat="51.5173639" lon="-  
0.140043" version="1" changeset="203496" user="80n"  
uid="1238" visible="true"  
timestamp="2007-01-28T11:40:26Z">  
  <tag k="highway" v="traffic_signals"/>  
</node>
```

Figure 2.1: *Example of a Node represented as an XML element*

Ways represent any linear characteristic of a map and are defined by an ID and an ordered list of one or more Nodes. Ways must be continuous and non-branching and can have different properties depending on the tags associated with them. Therefore, Ways can map many different characteristics of a map, from roads and rivers to buildings and areas. A distinction is made between open and closed Ways. Open Ways must have different first and last Nodes and usually map features such as roads and railway lines. All open Ways have a direction even if the feature mapped has no inherent direction (for example, a wall). Meanwhile, closed Ways must have the last Node identical to the first Node and usually map

features such as barriers, plazas and buildings (71).

```
<way id="5090250" visible="true"
timestamp="2009-01-19T19:07:25Z" version="8"
changeset="816806" user="Blumpsy" uid="64226">
  <nd ref="822403"/>
  <nd ref="333725781"/>
  <nd ref="333725774"/>
  <nd ref="333725776"/>
  <nd ref="823771"/>
  <tag k="highway" v="residential"/>
  <tag k="name" v="Clipstone Street"/>
  <tag k="oneway" v="yes"/>
</way>
```

Figure 2.2: *Example of a Way represented as an XML element*

Relations represent a group of Elements that behave like a singular entity and are defined by an ID, one or more tags and an ordered list of other Elements. They are used to represent logic or geographic relationships between objects, such as bus and trekking routes, and boundaries of areas like pedestrian zones or city/region confines (72).

```
<relation id="10952412" version="1"
timestamp="2020-04-02T21:50:55Z" changeset="83001051"
uid="202726" user="whb">
  <member type="way" ref="711863509" role="from"/>
  <member type="node" ref="6222512977" role="via"/>
  <member type="way" ref="664783357" role="to"/>
  <tag k="restriction" v="only_straight_on"/>
  <tag k="type" v="restriction"/>
</relation>
```

Figure 2.3: *Example of a Relation represented as an XML element*

The OSM project uses tags to represent the attributes of the various objects in the world. Each tag is stored in the database as a *key=value* pair, and both parts are text fields without format restrictions. Keys are used to describe topics, categories or general features of Elements. They can also be augmented with prefixes or suffixes, adding additional information to other tags in the same Element or forming namespaces. Typical uses of namespaces are language-specific tags, such as monument names in different languages or date-related specifications. Each Key requires a mandatory Value, which is also represented as a text field without any restrictions. Therefore, Values can be made up of distinct Elements or groups of Values, which can also be string literals such as names or numerical values such as distance values. Where a tag is absent from an Element, it inherits it from the parent, or if absent, there are agreed upon default values. Although conventions for most objects and features are agreed upon and present on the documentation, the free-form nature of both text fields can and often leads to differences in tagging styles between regions (73).

2.2.3 Reliability and accuracy concerns

Many scholars have used OSM to tackle different research projects as it is more accessible to licence and can potentially be more detailed than the licensed alternatives ((74),(75)). The heterogeneity and lax approach to tagging of OpenStreetMaps has made it easy for new users to contribute by mapping their local areas. However, it has also made developers sceptical about the data's usability due to the volunteers' different levels of knowledge and ability (76). Because of this, much analysis has been put into estimating the reliability, accuracy and completeness of the OSM database ((77),(78),(79)).

The overall consensus is that the OSM dataset can be highly complete and accurate, but it varies widely between world regions. For example, in Germany, the OSM representations of large cities are often more accurate than administrative datasets, but the same does not apply to rural areas or other countries (80). Therefore, any specialised router will have to deal with the incompleteness of the information (81).

Another potential problem that arises from the lax approach to tagging is the lack of standards between the OSM volunteers. This leads to each region developing their own *tag dialects* and forcing researchers to tailor their application to the region they evaluate. One of the most prominent examples of this behaviour is the mapping of sidewalks, where there are two accepted ways of mapping them at the time of writing. One approach adds a "*sidewalk*" tag to the road that features the sidewalk, while the other creates a separate Way in OSM to represent the sidewalk independently (82).

Therefore, each region needs to be independently adapted to create a generalised application that can operate globally. A saving grace is that the community promotes redundancy instead of contradictions. This means that it is unlikely for two tagging dialects to contradict each other's information, making the process of importing a specific region's data a matter of parsing information using the local standards.

Chapter 3

State of the Art

Routing pedestrians without disabilities is already a more difficult task than routing cars due to the safety requirement of the users and the need to stick to pedestrian ways as much as possible. Furthermore, it becomes even more of a problem when the routing algorithms need to consider accessibility barriers in the environment. Information about the obstacles and facilities is not always readily available. Even when there is a high level of information density, there could always be undetected or temporary barriers that could seriously impact the user by preventing them from reaching their destination. Because of this, people with disabilities are wary of one-size-fits-all solutions, often sticking to already known routes and planning new courses in advance of the trip itself.

Therefore, there is a need for an accessibility assistant rather than a routing tool. The unavoidable unreliability of the data coupled with the unknown completeness of it cannot guarantee a safe route for mobility-impaired pedestrians. Moreover, they need to be as much part of the process of generating a route as the algorithm. Only when provided with enough information will a user be sure about the accessibility of a given path. This has been a focus of many researchers in recent years, and this chapter aims to showcase many approaches to this problem that are worthy of note.

This chapter starts with an overview of the various approaches, summarized in

two tables. The first one compares the various features offered by them. Then, a taxonomy of the accessibility requirements is outlined, and then the approaches are compared with each other on which accessibility requirements they tackle. After the overview, two of the most comprehensive approaches are explored in detail. Next, different approaches are covered based on the data source they use for their routing. Finally, other approaches worthy of note but not close to the research question are briefly covered.

3.1 Overview

The table 3.1 lists all of the research about to be discussed in the following sections and outlines the features they offer. The target column lists the various application’s end-users between general pedestrians, mobility-impaired people and wheelchair users. The dataset column indicates the source of the data that the application uses. The real-time column indicates if the application has a low response time and can be used in real-time; if not, they were offline surveys or computationally heavy applications. The exposed information column indicates if the application discloses the accessibility information of the route to the user. Finally, the user input column indicates if the user has any control over the generated route after inserting their preferences, such as banning certain routes or querying for alternative paths.

Name	Target	Dataset	Real-Time	Exposed information	User input
Neis2015 (38)	Wheelchair Users	OSM			
U-Access (39)	Mobility Impaired	Private	X		
Hashemi2017 (40)	Wheelchair Users	OSM	X		
Kada2006 (83)	Pedestrians	Private			
OurWay (84)	Mobility Impaired	OSM + Other	X		
RouteCheckr (85)	Mobility Impaired	Private	X	X	
Path2.0 (86)	Mobility Impaired	OSM + Other	X		
Sasaki2013 (87)	Pedestrians	OSM	X		
mPass (88)	Mobility Impaired	OSM + Other	X	X	X
Quercia2014 (89)	Pedestrians	OSM + Other	X		
PAM (90)	Mobility Impaired	Private	X		
Mancini2014 (91)	Mobility Impaired	OSM	X		
Kulakov2015 (92)	Mobility Impaired	OSM	X	X	
CAPRA (93)	Wheelchair Users	OSM + Other			X
MobiliSIG (94)	Wheelchair Users	OSM	X	X	
Route4U (95)	Mobility Impaired	OSM + Other	X		
Routino (96)	Pedestrians	OSM	X		
ORS (97)	Pedestrians	OSM	X		X
Google Maps (98)	Pedestrians	Private	X		X
ARPA	Mobility Impaired	OSM	X	X	X

Table 3.1: A table comparing the different features offered by the approaches covered in the state of art

The table 3.2 covers the accessibility features that each of the approaches takes into account. Each of the columns represents a set of the accessibility requirements explored in section 2.1.4, and they are grouped as follows:

- The "*Path width*" column outlines if the width of the road is taken into account in the routing.
- The "*Stairs*" column outlines if there is an option to avoid stairs or to require certain features such as ramps or handrails for them to be usable.
- The "*Slopes*" column outlines if the road incline or camber is taken into account.
- The "*Surface type*" column indicates whether the road surface and conditions are taken into account.
- The "*Road Type*" column indicates whether the routing algorithm tries to feature specific roads such as sidewalks and pedestrian paths.
- The "*Crossing Type*" column indicates whether the routing algorithm tries to feature specific road crossing layouts such as supervised crossings and pedestrian islands on larger roads.
- The "*Kerb Height*" column represents if the algorithm regards the height of the kerb and access modality to the sidewalk.
- The "*Walkability*" column groups the more abstract preferences and requirements such as noisiness, beauty, and general preferences such as the presence of accessible facilities, level of foot/bicycle/car traffic and street furniture.

Furthermore, there is a distinction in how the requirements were tackled in the various research papers. For example, some applications defined the accessibility parameters as constants and only checked for their existence when computing the route. Instead, others described them as user-defined variables, allowing more granular accessibility settings to represent a user's preferences better.

Legend								
X	Takes into consideration the requirement as a constant, only checking for its existence							
D	Takes into consideration the requirement as a user-defined value.							
Name	Path width	Stairs	Slopes	Surface type	Road type	Crossing type	Kerb height	Walkability
Neis2015 (38)	X	X	X	X	X	X	X	
U-Access (39)	X	X	X				X	X
Hashemi2017 (40)	X		X	X				X
Kada2006 (83)								
OurWay (84)								X
RouteCheckr (85)			X		X		X	X
Path2.0 (86)								
Sasaki2013 (87)	X	X	X		X			X
mPass (88)	X	X		X	X	X		X
Quercia2014 (89)								X
PAM (90)	X	X	X	X		X		
Mancini2014 (91)	X		X	X			X	
Kulakov2015 (92)						X		X
CAPRA (93)			X					
MobiliSIG (94)	D		D	D				
Route4U (95)	D	X	D		X		D	
Routino (96)	X			X	X			
ORS (97)	D	X	D	D	X		D	
Google Maps (98)					X			
ARPA	D	D	D	X	X	X	X	X

Table 3.2: A table comparing the different accessibility requirements tackled by the approaches covered in the state of art

3.2 A review of ”*Collaborative personalized multi-criteria wayfinding for wheelchair users in outdoors*”

In their work, Hashemi and Karimi wanted to create an accessible wayfinding algorithm that combined a multi-criteria approach that considers the routing preferences of the user with a collaborative wayfinding approach that considers other users’ feedback (40). This approach is interesting to showcase because it uses a combination of collaborative crowdsourcing and VGI to tackle the shortcomings of both singular approaches.

Approach

They argued that databases of wayfinding systems have to contain specific attributes of sidewalks and routes to work correctly, which would not always be the case. Furthermore, they claimed that because collaborative wayfinding relies on feedback, it would require a considerable amount of users to protect against incorrect feedback. They explain that a combination of the two would use the strengths of each approach to reduce the faults in the other, making a better system overall. Moreover, they used a novel approach in gathering the user’s preferences, allowing users to compile a guided questionnaire instead of inserting numerical values, arguing that users may not correctly represent their impairments with values from 0 to 1 when asked.

Requirements

The requirements taken into account by Hashemi and Karimi are width, surface condition, elevation changes, sidewalk traffic and slope inclination. They chose this set of requirements based on the recommendations in the ADA standard (36). Each of these requirements has a weight assigned to them, generated with a hierarchical analytic process that extracts the weights from the questionnaire compiled by the user using a binary comparison matrix.

Implementation

As a preprocessing step, they represented the network of roads and sidewalks given by Open Street Maps as a graph. Then, from this graph, they pruned all the sidewalks narrower than 91.44cm as they would not be suitable for wheelchair users, as per the ADA recommendations. Next, they represented the elevation changes as the number of vertical changes on the sidewalk greater than 1.27cm. Following, the sidewalk condition was represented by the number of cracks on the sidewalk, the evenness of the surface, and utility holes on the road. Finally, the levels of traffic were represented by three distinct levels (heavy, light and no traffic) that were gathered based on field interviews conducted by them in previous research.

Using the user-generated weights, they assigned to each segment an accessibility index, which was then used as the weights in a weighted Dijkstra algorithm to find the best route. Once the path is generated, it is presented to the user as the best available route with no information about it and with no input from the user if not changing the previously defined weights. Furthermore, once a user completes their journey, they are asked if they believe that the route they have taken was the best possible road, with the choice of answering "yes", "no" and "do not know". If the user chooses yes or no, they will be further prompted to rate the confidence in their answer from 1 to 5. The answers to this are then translated into a mitigation coefficient that varies from 0.997 to 1.003, which affects the sensitivity of attributes in the routing of other users.

Evaluation

Hashemi and Karimi split the evaluation of their project into two different parts. A first evaluation was carried out on the routing algorithm without user feedback. Here they used a *Z-Statistic* that attributes a numeric value to the accessibility of a given route scaled up by its length. Then, they compared their algorithm against a route created with a fuzzy approach and the shortest path and found that their system provides a better accessibility score than the other algorithms. A second evaluation showcases the impact of user feedback on the routes. For this they used a Monte Carlo simulation repeated 100000 times to understand and test how the

algorithm changes, showing that the user’s feedback influences the routing based on the amount of positive and negative answers.

3.3 A review of ”*On Combining Crowdsourcing, Sensing and OpenData for an Accessible Smart City*”

In their work, Mirri et al. propose a system that combines crowdsourcing, crowd-sensing and open data to provide personalised paths to walking impaired people in an urban environment (88). This research is showcased as it is the closest approach to the outlined research question, as it also gives users information about the path they get proposed and it allows them to choose between a set of paths instead of a single one.

Approach

The system comprises two components: mPASS (mobile Pervasive Accessibility Social Sensing) gathers the data about the urban environment with crowdsensing and crowdsourcing, and WhenMyBus collects information about the city’s public transport such as accessibility equipment and real-time location. Users can input preferences about the various accessibility features and get a personalised route to their destination.

With mPASS, Mirri et al. combined various sources to get a comprehensive view of the urban environment. When the application runs on mobile devices, it uses the phone’s sensors to determine accessibility features in the environment. Then, it allows users to validate the automatically gathered features and add new features of their own. Furthermore, organisations responsible for urban accessibility can add official information and fix data collected by other users.

Instead, the WhenMyBus system uses open data shared by the bus operating companies to retrieve information about accessibility features, informing users

only of the busses that can safely carry them, and displays real-time data and timetables.

Requirements

The approach taken by Mirri et al. stores the various accessibility barriers and features into a set of broad categories. These categories are:

- **Gap**, used to identify accessibility features that obstruct the environment, such as gaps and steps
- **Cross**, used to identify characteristics that relate to crossing a street, such as traffic lights and zebra crossings
- **Obstruction**, used to identify all street furniture, such as trees and garbage bins
- **Parking**, used to specify accessible parking spaces
- **Surface**, used to describe road surfaces, such as material and maintenance levels
- **Pathways**, used to describe sidewalks and their characteristics
- **Bus stops**, used to describe bus stops and their characteristics
- **Bus**, used to describe the facilities and barriers, such as wheelchair anchorage and steps, of a specific bus

Implementation

When users first create their profile in the application, they are prompted to rate each category representing the requirements. For each category, users are able to rate it as "Neutral", "Like", "Dislike", and "Avoid".

To list accessibility features and barriers, the system uses geotags called aPOI (Accessibility Points of Interest) by Mirri et al. Each aPOI contains its category, a description, and trustworthiness level based on the number of reports about

it. aPOIs can be added into the application in various ways, which affects the type of aPOI and its trustworthiness. The sensor generated ones are the least reliable, user-added aPOIs are considered of mixed reliability, and organisation-added aPOIs are always regarded as accurate and trustworthy as only official organisations can add them.

When a user requests a route, the system generates a graph from the road network on OpenStreetMap. Then it removes all the roads that have aPOIs marked as "Avoid" by the user. Then, each road in the remaining graph is given a weight based on its length and the amount of aPOIs marked as "Like" and "Dislike". Finally, a Bidirectional Dijkstra algorithm generates a set of paths, with the number of proposals also defined by the user. For each path, the user is able to explore all the aPOIs that they will potentially encounter.

Evaluation

Mirri et al. showcased a case study that highlighted the differences of a route proposed with their system against an unnamed traditional mapping platform to evaluate their system. The comparison showcased how their approach avoided routing the pedestrian in a path with accessibility limitation and the fact that it took into account the increased walking time of a person with walking impairments in their time estimation. Furthermore, when suggesting to the user to take public transport, their system listed the accessibility features of the bus.

3.4 Commercial applications

Other interesting implementations of pedestrian-focused routing engines fall among the publicly available and commercial ones. For example, among the most popular ones, we can find Google Maps. Following, Routino and OpenStreetMap are two of the most used open-source routing engines. And finally, the crowdsourcing application Route4U, which focuses on promoting accessibility in various cities.

Google Maps

Google Maps is a consumer mapping service provided by Google, and it offers real-time route planning for users travelling by car, foot, and public transport (98). Going into detail on the foot and public transport aspects, Google Maps does not provide any accessibility setting, only focusing on general pedestrian paths. It assumes that the walking speed of a pedestrian is 5km/h, and the only accessibility information it shows is the height changes along the route. When the user asks for a course, the service provides several potential paths that they can choose from. The data it uses is privately gathered by Google, and the routing algorithm used is also undisclosed.

Route4U

Route4U is a consumer application developed by a company under the same name (95). It uses a mix of open data and official information provided by partnered cities to map the sidewalk networks. Because it uses official data, the application's coverage is limited to the cities partnered with the initiative. It allows users to define a small set of settings that reflect their walking abilities. These settings include the maximum and minimum kerb height, the maximum slope inclination and the minimum width of a sidewalk. When asked for directions, the application shows a single best path without giving any accessibility information about the road. Another feature of the application is that it crowdsources accessibility barriers, which are displayed to users when a route goes through them. The routing algorithm behind the application is also undisclosed.

Routino

Routino is a generic routing engine developed in C that uses OpenStreetMap data as the base information for its road networks (96). As a first one-time step, it pre-processes the OpenStreetMap data into a custom database. This custom database stops representing the OSM data, changing it to remove unnecessary complexity, thus increasing the algorithm's speed. Being a routing engine instead of an application, Routino does not have defined parameters for vehicles or pedestrians but rather allows for varied customization options in the form of profiles, which map

to a set of weights for many possible road features. These preferences include, but are not limited to, the type of street, the road surface and specific height and width requirements. The routing algorithm is an A* algorithm that finds the shortest route.

OpenRouteService

OpenRouteService is a suite of open-source geographical tools maintained by the Heidelberg Institute for Geoinformation Technology (97). All the tools use OpenStreetMap as the primary data source. Some of the tools they provide are their direction service APIs, which allow users to get a route between two points. The most related to this dissertation are the pedestrian and wheelchair direction APIs. The pedestrian API gives a course that tries to maximise the use of sidewalks, allowing users to mark if they would like to avoid stairs. The wheelchair API is more complex, allowing for more detailed accessibility requirements to be specified, but it is also limited only to zones that have enough accessibility information on OpenStreetMap. The application gives no accessibility information about the course it generates, but it allows users to query for up to 3 alternative paths. The algorithm used to calculate the route is an implementation of the hybrid C-ALT algorithm ((99),(100)).

3.5 Routing based on privately gathered data

Scaling down the problem to a smaller area allows researchers to use privately gathered geographic information systems, carry out field studies to collect missing accessibility information, and manually evaluate the application. In the following section, a selection of such approaches is highlighted.

RouteCheckr

In their work on RouteCheckr, Völkel and Weber developed a client/server architecture that would act as an accessibility tool for the Technical University of Dresden (85). The application is made of a routing component that creates paths based on the walking ability of the user and an annotation component for temporal

obstructions and additional information. They gathered the requirements for their application from a survey with 88 visually impaired respondents and a literature review. They manually collected the road and sidewalk network data from the college grounds, which they then used as the basis for their tool. Each path in the college ground has been given a safety rating for each mobility-impaired pedestrian group, based on the collected data. User annotations can further map this data out. The annotations are implemented as multimodal annotations that consider the user's geographical information, impairment category, and LOM-Modality, which combines the user's location, movement, and orientation. The algorithm used for the routing is an implementation of the Dijkstra algorithm modified to allow routing. The user is limited in their accessibility selection, only selecting their group and weight preference between security and length. Only a single route is presented to the user, and they are able to see the security index and annotations for all stretches of the road. Finally, they utilised their application to analyse the relation between user profile, annotation data and resulting generated route to evaluate their work. They found that the outputs adequately reflected their expectations in the tradeoffs between the route length and the user-defined criteria.

U-Access

U-Access was a routing web app developed by Adam D. Sobek and Harvey J. Miller that let users obtain pedestrian-focused directions, offering the option to choose between three levels of walking ability (39). They used data from a private GIS and integrated it with accessibility information obtained from The University of Utah (USA), where they also did the evaluation. To create the sidewalk network, they assigned a rating from 1 to 3 to each node, where 1 represents a street with non-accessible features, 2 a street accessible by users with aided mobility and 3 a street accessible by wheelchair users. Then it creates three distinct graphs for each of the three types of users. Once the graphs are created, it uses a Dijkstra algorithm to find the shortest path between the start/destination point on the graph corresponding to the walking ability defined by the user. The user cannot define or choose anything other than select their walking ability between the three choices, and they are not shown any accessibility information about the path.

To evaluate the application, they gave it to the Center for Disability Services of the university, which made it available to students and staff for three weeks. Afterwards, the test users were given a questionnaire, which gave favourable results overall.

PAM

Hassan A. Karimi, Lei Zhang and Jessica G. Benner proposed a definition for personalised accessible maps (PAM), also showing a prototype carried out in the university of Pittsburg called PAM-Pitt (90). The set of requirements for mobility-impaired pedestrians were gathered with an extensive literature review. The information about the campus grounds' pedestrian network was given to the researchers by the university, along with the location of buildings, facilities, and campus entrances. A novel approach to providing directions has been introduced; because 20% of the campus roads did not have names, they used landmarks to give directions to specific categories of users. Users can select from some predefined impairment categories, and they will get a single best route based on that, with no additional accessibility information displayed. Furthermore, the routing for wheelchair users is done with a fuzzy approach. Finally, because the research acted as a proof of concept rather than a formal evaluation, the researchers outlined the challenges they faced in the development, describing the technological limitations that need to be overcome before creating a working application.

3.6 Routing based on VGI

With the popularisation of the Internet, Volunteered Geographic Information became a widespread alternative to privately gathered GIS. This allowed the crowdsourcing of accessibility information in cities, and researchers have tried different approaches to create routing systems that use this open-source dataset. In this section, three of the most interesting methods are showcased.

”Walking Route Recommender System Considering SAW Criteria”

W. Sasaki and Y. Takama proposed a walking route recommender system based on SAW criteria: safety, amenity, and walkability (87). The system was designed to promote walking as a healthy habit, and because of this, the recommender creates circular routes that start and end at the same user-defined location. Users can specify the length of the journey, any particular waypoints they would wish to pass and are given 10 *points* to assign as weights between the three criteria, which affect the generated route. To create the road network, they translate the data gathered from OpenStreetMap into RDF stored into a SPARQL database, then queried during the route generation. The route generation is done using the A* algorithm. Despite considering many accessibility features, the user can only define their preference between the three criteria and is provided with no accessibility information about the path the algorithm generates. To evaluate their system, five volunteers created routes by defining the weights and then walked the generated paths. After they finished the circuit, they were asked to rate it according to the previously specified three criteria. In most cases, the volunteers gave good scores to the criteria they put as high priority meaning that the recommendations were adequate.

”Point to point navigation for people with mobility impairments”

A. Mancini and P. Zingaretti introduced a mix between hardware and software to provide navigation to mobility-impaired people (91). In their approach, two graphs are created which differ in the level of detail. The less detailed graph maps only the road networks and is utilised for travelling with cars or bikes. Meanwhile, the more detailed one routes pedestrians by mapping the sidewalk network with the related accessibility information when available. The A* algorithm is applied to find the shortest path in both graphs. Users are not be able to insert any accessibility preference and are not given information about the accessibility of the proposed path. The novel ideas presented in their research are the use of text to speech to give directions to visually impaired users and a wearable device that uses a Laser Range Finder to identify obstacles. When an obstruction is detected, it reroutes the user to a different path.

”Measuring the reliability of wheelchair user route planning based on volunteered geographic information”

To measure the reliability of Volunteered Geographic Information, P. Neis introduced a personalised routing algorithm for wheelchair users (38). Building on his previous research (101) and literature review, he assigned a set of weights to various road features. Then, user-defined weights were combined with these constants to create an impedance score for each node in the graph. This score was then multiplied with the length of the segment to generate the cost of each connection in the graph. As this algorithm was developed as a survey of OSM data, there is no user customisation other than the weights, and the route provided features with no accessibility information. To test the robustness of this approach, they introduced a reliability factor to the route computation, which calculated how many of the segments of the route contained the desired accessibility information in the OpenStreetMap database. They then tested their approach on the area of Bonn, Germany, to discover that the information was enough to create accessible routes reliably.

3.7 Other approaches

Other approaches have been explored to find a solution for routing walking impaired people or improving the quality of the routes generated for pedestrians of all types. Now, A selection of approaches that were not close enough to the research question or slightly different from other methods already showcased is discussed. One interesting approach was that of Ghaberachi et al. (94) which instead of focusing on a strict set of requirements they asked the users to fill a questionnaire about the user’s confidence level towards various road features, which then acted as the weights in a fuzzy approach. Kulakov et al. (92) built upon previous research by creating an OpenStreetMap-based routing algorithm that provides a mobile application that allowed users to get directions in real-time. Furthermore, in his work on CAPRA, Rahman et al. (93) approached the problem by focusing on height differences creating a contour-based pathfinding algorithm. Palazzi et al. (86) took on a fully crowdsourced solution that used the GPS data of users to

provide directions to other users with similar impairments, going off the assumption that paths frequently utilized are also the most accessible ones. With another crowdsourced solution, Holone et al. (84) tasked the users with manually rating the accessibility of each road segment, using that data to provide routing directions.

Focusing more on general pedestrians rather than mobility-impaired people leads to different approaches to increasing the wellbeing of citizens through walking. For example, Quercia et al. (89) proposed a routing algorithm that routes pedestrians to safe, quiet and beautiful streets by scraping social media posts and crowdsourcing voting on vague roads. Kada et al. (83) instead focused on finding a reliable data source for the routing, developing an algorithm that takes as an input raster maps of neighbourhoods.

As we can see, there has been significant research put into routing for pedestrians. However, the main factors that have not enabled widespread utilization have been an incomplete data source and the overreliance on crowdsourcing with undefined standards, leading to regional differences in how data are annotated and stored and thus preventing the creation of a one size fits all solution.

Chapter 4

Design and Implementation

This chapter starts with an overview of the project, detailing the functional architecture, the technologies used and the general flow. Then the three central processes are described in detail, starting with the gathering of the input, then explaining the creation of the graph, and concluding with the routing algorithm.

4.1 Project overview

This section outlines the functional architecture of the application, summarises the application's flow when a user requests a route and reports the technologies used to implement the design.

4.1.1 Functional architecture

The application has been developed using a functional programming paradigm. In functional programming, the code is developed using *pure* functions rather than classes. A pure function is any function that is not influenced by outside scope, in which the same output will always be returned given the same input. This paradigm allows for clean, clear and maintainable code as not having a state reduces the possible side-effects in the application.

Therefore, the components are a representation of collections of functions rather than traditional classes. The system comprises of four main components that han-

dle all of the application's functionalities: the Web service, the Geocoder, the Graph handler, and the Routing handler.

Web service

The Web service handles the frontend of the application and manages the general flow. The user will interact with web pages to insert their preferences, input the start and end points, and see a visual representation of the generated directions. Because this component is a wrapper for the more complex components used in the routing, it is trivial in its implementation and will not be explained in detail in this chapter.

Geocoder

The Geocoder handles the external API calls to transform the text descriptions of the start and end locations given by the user into OSM data Elements.

Graph handler

The Graph handler parses the OSM data and creates a representation of the road network the user is interested in as a graph while also collecting the accessibility information of each road.

Routing handler

The Routing handler gathers the road network graph, the user input and their preferences. Then, it uses an A* algorithm to find the most accessible route that connects the user-inserted start and end locations in the graph respecting the user's preferences.

4.1.2 Application flow

In diagram 4.1 the general flow of the application is displayed. When users first open the application's web page, they will be prompted to insert their preferences

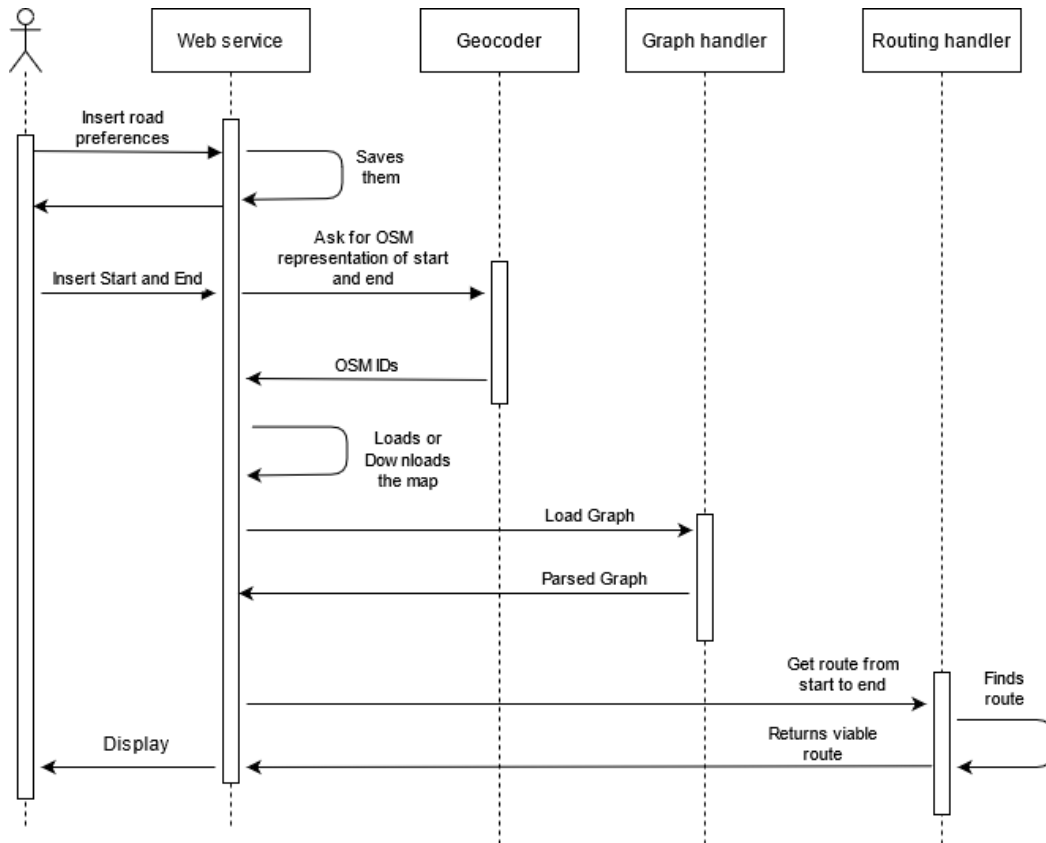


Figure 4.1: *Sequence Diagram* outlining the flow of the application when a user queries for directions

regarding various types of accessibility characteristics ¹. After entering their preferences, the users will be redirected to the application's main page, where they will insert the starting and ending point of their journey in a text field. These text values, which will likely be road addresses, are then translated into OSM Elements using the Geocoder. Along with obtaining the OSM representation of both places, the application also obtains their coordinates. These coordinates will be used to create a bounding box that will act as the interesting road network for the routing, and all information inside it will be downloaded from OSM.

Once the data is downloaded, the Graph handler will parse it, creating a graph representation of the road network and gathering all the available accessibility information. The generated graph is then cached so that future requests in the same area will not need to be downloaded and parsed again. Afterwards, the Routing handler runs an A* algorithm on the graph, using the accessibility preferences of the user as modifiers of the cost function. Finally, after the routing handler finds a suitable route, it is displayed to the user along with all the gathered accessibility information for each step of the road ². Here the user will be able to ban any particular step of the displayed directions that they do not deem adequate and get delivered another route that avoids that road.

4.1.3 Technology choices

The programming language chosen for the application is Python (102). This choice was made as it is a flexible programming language that allows for quick prototyping. The downside of Python's flexibility manifests with the slower computation times. However, the slower times are still satisfactory in regards to the scope of this dissertation.

Flask (103) is the web framework used to create the web service at the base of the application. Furthermore, *Jinja* (104) was used as a templating engine to render the various web pages. Both of these technologies have been chosen because

¹A screenshot of this step can be found in Appendix A.2

²A screenshot of this step can be found in Appendix A.1

of their ease of integration with Python.

Nominatim (105) is an open-source geocoder developed as a search engine for OpenStreetMap and is the officially supported tool by the project. Many different geocoding applications exist, but because of its close integration with OSM, only Nominatim returns the IDs of the OSM Elements in its responses, which is essential in the current version of the application. The OSM project provides a public API to access Nominatim, but because its website uses the same API, it is rate limited to only two queries every second. Because of this, the dissertation uses a combination of the public API and an additional private API on RapidAPI (106), a platform in which users can host and manage APIs on cloud environments. Moreover, Nominatim presents some drawbacks in its geocoding functionalities. For example, when multiple options are found for the same description, Nominatim returns the single feature it deems most important. Because of this, on ambiguous queries, the Geocoder may return results that are not relevant, and this behaviour cannot be programmatically managed.

To download the map information from the OSM database, the Overpass API (107) is used. Overpass is an open-source service with a public API used to download custom selected parts of the OSM database in an XML format.

4.2 Gathering the input

This section explains the processes that are executed to gather the input for the application. First, the weights used in the routing are defined, and an explanation of how the user inputs them is given. Then, the interactions with the geocoder are described. Finally, an explanation of how the raw map data is downloaded from OSM is given.

4.2.1 User weights

The user preferences are treated as weights in the routing algorithm and are represented as a dictionary of decimal values between 0 and 1. The user will be able to

set these preferences either using a questionnaire or by manually setting the values. The questionnaire comprises eight questions that map to the different accessibility requirements tackled by the application. The 4.1 table outlines the questions that are used in the questionnaire.

Question	Description	Accessibility Category
How important is avoiding routes shared with bicycles?	Routes the user avoiding roads that might be shared with bicycles	Walkability
How important is going on lit roads?	Routes the user toward roads with illumination	Walkability
How important is going on a road with a good surface material?	Routes the user towards roads paved with wheelchair-suitable materials and in good conditions	Surface type
How important are short kerbs?	Routes the user towards roads with low kerbs or kerbs accessible with a sloped incline	Kerb height
How important are supervised crossings?	Routes the user towards supervised street crossings where possible	Crossing type
How important is the width of a road?	Routes the user towards roads with wider than a user-defined value	Path width
How important is the incline of a road?	Routes the user towards roads with a lesser incline than a user-defined value	Slopes
How important is avoiding stairs?	Routes the user avoiding stairs, except in special cases defined later	Stairs

Table 4.1: *A table outlining the questions used to gather the user’s preferences towards the different accessibility features*

For all the questions, the user will be able to reply by either manually inputting a decimal value or by choosing one of 5 answers, which will set decimal value for the user. Table 4.2 outlines the possible responses.

Furthermore, as the application is a pedestrian-focused router, the default is

Response	Not Important	A little Important	Important	Very Important	Essential
Value	0	0.25	0.5	0.75	1

Table 4.2: *A table outlining the decimal equivalent of the possible responses to the questions*

routing the users towards pedestrian roads or streets with sidewalks. This is achieved by setting an hidden preference for sidewalks as "Essential".

As previously mentioned, for some of the questions there are additional user-set parameters. For example, in the case of the path width and the inclination, the user will be able to set their preferred value. Furthermore, in the case of stairs, the user will be able to select the option of using stairs if they present a handrail, ramp, or if below a number of user-defined steps.

4.2.2 Geocoder

After the user has entered their preferences for the road features, they will need to input their journey's start and end destination. They will be able to do so with two text boxes where they will potentially enter a road address or the name of a place/building followed by the city name. As a first step of the routing process, the application needs to determine the position on the earth's surface of this text description of a location.

It does this using a Geocoder, a tool that will return a set of geographic coordinates given a description of a location. Many such tools exist as commercial APIs, but this application will use a particular version compatible with the OSM database called Nominatim, introduced in section 4.1.3. Along with the coordinates, Nominatim will also return the OSM element that closest represents the desired location.

The application will execute two queries to Nominatim's API. The first query will use the text description given by the user, which will return an OSM element

that can be either a Node, a Way or a Relation. This element's type, ID and coordinates are saved, and a second query is then executed. Because there is no requirement for any given OSM element to be associated with a road, the closest road also needs to be discovered as a starting point for the algorithm. Therefore, the second query will satisfy this requirement, searching for the nearest road to the first element by using the element's coordinates. Once both queries are executed, the location will be represented by the location's OSM type, ID and coordinates, and the road ID that is closest to it.

This process is repeated for both the start and end coordinates before proceeding further.

4.2.3 Map data

After obtaining the coordinates of both start and end, the application will need to get the road network that links the two coordinates together. To achieve this, a bounding box will be created around the two coordinates. Furthermore, the borders will be larger than necessary to avoid the edge case of having all possible connections going through the outside the bounding box and to capture more future queries as this map will also be cached.

If the application finds that the map that included both coordinates has already been downloaded, it will load the previous cache to avoid redownloading files with potentially large sizes.

An API is used to download all the information inside this bounding box, which will be saved in an XML format. For further discussion about the API, refer to section 4.1.3. After the data is downloaded, it is saved to memory for future use and then transferred to the Graph Handler for processing.

Caching the raw XML file has been done for development purposes. However, in a finalised application, this step might be skipped in favour of only caching the processed graph to save disk space.

4.3 Creating the graph

This section explains the processes that are performed to create the graph representation of the road network. First, an explanation is given on how the application parses the OSM Data. Then, the graph structure is explained. Finally, an explanation of how the accessibility information is extracted from the road network is given.

4.3.1 Parsing the data

After gathering the XML file, the application will parse it to create a graph. Each of the three types of Element outlined in section 2.2.2 will need to be parsed independently to create the objects required to make the final graph.

The first step will be parsing all Nodes to gather their coordinates and accessibility information in a dictionary. The coordinates will be used to determine the distance between points in the routing algorithm, while the accessibility information will be used to enhance the accessibility information gathered from the Ways. Which accessibility information is extracted from the nodes is discussed in section 4.3.3.

Afterwards, the application will parse all the interesting Relations. From the variety of concepts represented by Relations, the interesting ones are the ones that represent plazas and pedestrian areas, depicted in OSM with the tag "highway=pedestrian" (108). These plazas are drawn with one or more referenced Ways that make up their perimeter, and the Ways do not have any requirement of being tagged in any particular manner, so they may not be marked as "highway" themselves. Because routing in open areas only described by their perimeters is problematic (109), the application will route users along a plaza's perimeter to cross them instead. Because of this, these potentially untagged Ways need to be added to the graph, so they are saved in a list that will be used later to add them to the graph.

After having parsed the Nodes and the Relations, the application will parse all the interesting Ways. From the variety of concepts represented by Ways, the interesting ones for a routing application are the ones that represent roads, depicted in OSM with the tag "highway" (110). Along with roads tagged with highway, the application will also parse all the Ways flagged as the perimeters of the pedestrian areas found while parsing the Relations. While parsing each Way, all the accessibility information related to the Way will be saved in a dictionary along with all the referenced nodes that make them up. Furthermore, the relations between Nodes and Ways will also be saved in another dictionary to track which Nodes are referenced by which Ways. Which accessibility information is extracted from the Ways is discussed in section 4.3.3.

The redundancy in the saved dictionaries, such as in the two different dictionaries for Ways-to-Nodes and Nodes-to-Ways references, has been created as a tradeoff between computational power and memory. Because many operations require both of the connections, recreating them by searching a single dictionary would rapidly increase the complexity of the functions, potentially slowing down the application.

Before creating the graph with the information parsed, the application will remove all of the useless nodes referenced in the ways so that the final graph will be of a smaller size. The Nodes that make up a Way might be intersections, features, and points used to represent the road's shape better. However, to create a suitable graph for routing, only intersections are needed. An intersection is characterised by a node being referenced in two or more Ways. Therefore, almost all of the nodes that are referenced by only one Way will be deleted. Exceptions will be made for the Nodes that feature as the first or last referenced node in a Way. Because the referenced Nodes in a Way are ordered, the first and last referenced Nodes represent the start and end of a road, respectively. These nodes might not be attached to another road (such as in cul-de-sacs) but are still relevant for routing.

Once the parsing is over, two objects will be used to create the graph: A dictionary of nodes with their corresponding coordinates and accessibility infor-

mation, and a dictionary of Ways with the associated accessibility information and referenced Nodes.

4.3.2 Creating the graph

All the Ways included in the Way dictionary are parsed to create a graph. Then, for each Way, all of the referenced nodes are analysed. Every node encountered will act as a node in the graph, using its ID as a dictionary key and an array of edges containing said node as the value. Because the nodes are ordered, the neighbouring nodes in the array are also neighbouring in real life. So, for each adjacent pair of nodes, an edge is added to the graph. This edge will have a cost equal to the distance between the coordinates of the two nodes and the Way's ID as a piece of additional information that will be later used to calculate the accessibility cost.

Once all the Ways have been parsed, the graph that represents the road network will be finished. Lastly, it is saved to disk and cached so future queries in the same region will skip the parsing operation.

This graph dictionary, along with the two dictionaries containing Ways and Nodes with their respective accessibility information, will be used in the routing algorithm.

4.3.3 Gathering accessibility information

When parsing each Element, its tags are scanned for accessibility information. The tags and corresponding values have been gathered from research on the OSM wiki, spot research in different geographic areas and analysis of the most common key/value pairs. Because OSM values are defined by the volunteer who mapped them, there is no certainty that the values will be standard across all the data. Therefore, while parsing, each value is mapped to a standard set of values before saving. Furthermore, as any category could be missing from the data, a value of "None" will always be admissible to mark a particular feature as unknown.

Nodes

Nodes map to singular features of the real world and are commonly used in OSM to represent kerbs and crossings when in conjunction with roads. Therefore, this is the accessibility information that is useful to gather from them. In table 4.3 the tags used to identify each characteristic are outlined, along with the values that are then saved.

In the kerb's case, only the three values "raised", "flush", and "lowered" are saved. "Raised" kerbs are those higher than 3cm, "lowered" kerbs are raised kerbs that provide an access ramp to the sidewalk, and "flush" kerbs are the same height as the street. These three values have been chosen as they are the most commonly agreed and popular values for kerbs (111). Furthermore, all the values that fall outside these three types will be mapped as "raised" as a default value.

The crossing information is handled in a similar way to kerbs. However, here the values saved are "unmarked", "unsupervised", and "supervised". "Unmarked" are spots in a road where it is possible to cross but without any road markings, "unsupervised" are spots where road marking is present but without any traffic signal, and "supervised" are crossings with markings and traffic signals. These three values are chosen as a simplification of the multiple types of crossings found in the OSM database (112), so when checking for crossing tags, various values are checked and then classified as one of these three values.

Ways

Many different accessibility features can be determined from the tags that are associated with a Way. Although Ways can represent various features in OSM, the Ways gathered through the parsing of the raw XML file all represent roads. The accessibility features collected are illustrated in table 4.4, along with the tags that identify said features and the saved values.

The first feature identified is whether the road analysed is a pedestrian road

Category	Description	Tags queried	Values saved
Kerb Height	Determines if a node represents a kerb, and if so its type and therefore height	barrier=kerb — kerb=*	None ——— raised ——— lowered ——— flush
Crossing Type	Determines if a node represents a crossing, and if so its type	highway=crossing crossing=*	None ——— unmarked — unsupervised — supervised

Table 4.3: *A table outlining the accessibility features gathered from Nodes, with the queried tags and the values saved*

or has a sidewalk. Because there is not an agreed standard on how pedestrian roads are defined, many different tags can be checked to determine this information (82). An explanation of this redundancy in tags is covered in section 2.2.3. After reviewing the various tags, a boolean value will be saved to represent this information.

The width of a road and its incline are standardised in how they are tagged, both having a single widespread key label, but can widely differ in the values entered by the volunteers. The OSM recommended unit for width is meters but can be represented as any unit as long as specified in the value field (113). Nonetheless, it will be saved as meters in the application. Furthermore, the recommended unit to represent incline values is the "rise over run" percentage, which is also how it is saved in the application (114). When volunteers are not sure about the incline amount of a road, they are recommended to insert "up" or "down" as the values. These values will be translated to an incline of 6% by the application before saving. This default value has been chosen because it is the upper recommended limit for wheelchair users, and, presumably, values above this limit would be considered abnormal and therefore tagged with the correct value.

To determine if a road is shared with cyclists, several different keys need to be analysed (115). First, there is a need to assess if cyclists can use the road or not. This is always true if the road is tagged as a cycleway, it is assumed true for

all other pedestrian roads unless specified otherwise. Then, a check is made for the "segregated" tag, as it defines if the pedestrian and cyclist lanes are shared or not. If it is determined that cyclists can use the road and are not segregated to a different lane, the value saved will be True.

Determining if a road has lighting is straightforward, with only one tag to check with "yes" and "no" as the most widespread values (116). These values will be saved as the respective boolean values.

There are three different pieces of information relating to the surface of a given road (117). The first is the surface material, the second is its smoothness, and the third is the "track type", which measures how well maintained it is. All three have popular standards that clearly define the values that volunteers should enter, and because of this, the values are saved as they are detected.

Lastly, five different tags are used to gather information about stairs (118). The first tag is the key/value pair "highway=steps", which is the agreed standard to identify a set of stairs. Furthermore, additional information on a set of stairs can be identified with the tags "handrail" and "ramp", both of which usually contain either the values "yes" and "no", or the location of the feature on the stairs (e.g. "left" or "right"). Moreover, the number of steps might be included under the "step_count" tag as an integer value. Finally, the presence of the tag "wheelchair" indicates if the volunteer has deemed the set of stairs wheelchair-accessible or not. The values about the presence of stairs, ramps, handrails and wheelchair accessibility will be saved as a boolean value, while the number of steps will be saved as an integer.

Category	Description	Tags queried	Values saved
Road type	Determines if a way has a sidewalk or is a pedestrian road	highway=* foot=* sidewalk=* segregated=yes	None True False
Path width	Determine the width of the way in meters	width=*	None [Float Value]
Slopes	Determine the incline of the way in "rise over run" percentage	incline=up incline=down incline=*	None [Float Value]
Walkability	Determines if the way is shared with bicycles	segregated=* highway=cycleway cycleway=* bicycle=*	None True False
Walkability	Determines if the way way has lighting	lit=*	None True False
Surface Type	Determine the surface of the way	surface=*	None [Text Value]
Surface Type	Determine the smoothness of the way	smoothness=*	None [Text Value]
Surface Type	Determine the track type (smoothness for dirt roads) of the way	tracktype=*	None [Text Value]
Stairs	Determine if the way represents a set of stairs	highway=steps	None True
Stairs	If the way represents stairs, checks if it has a handrail	handrail=*	None True False
Stairs	If the way represents stairs, checks if it has a ramp	ramp=*	None True False
Stairs	If the way represents stairs, checks if it is wheelchair accessible	wheelchair=*	None True False
Stairs	If the way represents stairs, checks the number of steps	step_count=*	None [Int Value]

Table 4.4: A table outlining the accessibility features gathered from Ways, with the queried tags and the values saved

4.4 Routing Algorithm

This section explains the processes that are performed to generate the optimal path in the graph. First, an explanation is given on how the start and end locations are dynamically added to the graph. Then, the costs which are used to determine the optimal path are defined. Finally, an explanation of how the path generated from the A* algorithm is encoded to a standard format is given.

4.4.1 Adding start and end points

At the start of the routing process, the application has the start and end locations with the associated roads and the graph representing the road network. However, because there is no certainty that the user inserted locations are part of the road network graph, as it has been previously processed and simplified, adding them manually is necessary so that the search algorithm will work properly.

In a process performed for both the start and end location, the location's coordinates are added to the Node coordinates dictionary. This is done so that it will be possible to calculate the geographical distance between the location and any other Node. Next, the location needs to be connected to the graph, so it will be connected with Nodes that reside on the same road. Therefore, all the Nodes that are part of said road in the graph are gathered, and in this group, the two geographically closest nodes to the location are determined. An edge is then added to the graph between these two Nodes and the user-defined location.

Once this is done for both the start and end locations, the graph will be ready for the search algorithm to search algorithm to find a path between the requested locations.

4.4.2 Routing algorithm

The routing algorithm used is A*, chosen because it offers a flexible approach that can efficiently adapt to any user requirement. The heuristic function used calculates the distance between the inspected node and the end node, which is

then multiplied by a constant³. The constant is used to reduce the number of nodes explored as it incentivises the algorithm to choose roads that lead to the destination even if they provide lower accessibility to the user. The value of the constant has been determined through testing.

A hop happens in the algorithm when the edge between two nodes is crossed in search of a path. As previously mentioned in section 4.3.2, each edge in the graph corresponds to an OSM Way. To determine the cost of a hop, three factors are considered. The first is the distance in kilometres between the two nodes. The second is a constant added to the cost when a hop uses an edge which Way is different to the one used to arrive at the previous node. This is done to incentivise the algorithm to use as few distinct Ways, and therefore roads, as possible in an effort to reduce the complexity of the generated directions. The third is the accessibility value of the Way used in the hop, calculated with an accessibility cost function.

The accessibility cost function calculates how close the accessibility features of the Way match the user's preferences; the higher the cost returned by the function, the more unsuitable the road is for the user. To calculate the accessibility cost, each accessibility feature is checked and assigned a sub-score based on its value, then multiplied by the user-defined weight. The sum of all the sub-scores is then divided by the sum of all weights to keep the scores contained even when multiple requirements are factored in.

Each sub-score is based on three different constant values: An "unknown" value, an "unsuitable" value and an "impossible" value. The sub-scores for the various accessibility features are handled as follows.

To calculate the walkability features' sub-scores, which determine if a road is lit or shared with cyclists, they are checked. If they present unfavourable values, the "unsuitable" score is assigned to them.

The values for the "width" and "incline" features are also checked to calcu-

³The value of all constants mentioned in this section can be found in appendix B

late the respective sub-scores. If the values are missing, the "unknown" score is assigned to them. Instead, if the values are present, they are compared against the user-defined values "min width" and "max incline". If they do not align with the user-defined values, the "impossible" score is assigned to them. The same happens for the "road type" feature. If the road type information is missing, the "unknown" cost is assigned. Instead, if the road is determined not suitable for pedestrians, the "impossible" cost is assigned.

If the road is determined to have stairs, the user's settings are compared against the stairs' features. If the user has specified that stairs can be traversed if they have handrails, ramps, or under a certain number of steps, the corresponding values will be checked. If any of the user preferences match, then the stairs will be ignored. Otherwise, a cost of "impossible" will be assigned to them.

If the value for the "surface type" information is present, the sub-score will be calculated using equation 4.2. The surface material, road condition, and track-type information are translated into decimal values using a set of constants⁴. The better the road conditions, the smaller the final score will be. Instead, if the surface value is not available, the "unknown" score will be assigned. A similar procedure is also performed for the "kerb height" and "crossing type" features, as seen in equations 4.3 and 4.4. The only difference can be found with the "crossing type" feature. Because a road segment might not necessitate a crossing, the "unknown" score is not assigned when the information is not found.

$$\text{Surface Subscore} = (1 - (\text{material} * \text{condition} * \text{tracktype})) * \text{unsuitable}$$

Figure 4.2: *Equation to determine the accessibility cost of the surface type of a Way*

⁴A table containing the decimal value representation of the various features can be found in appendix B

$$\text{Kerb Subscore} = (1 - (\text{kerbType})) * \text{unsuitable}$$

Figure 4.3: *Equation to determine the accessibility cost of the kerb height of a Way*

$$\text{Crossing Subscore} = (1 - (\text{crossingType})) * \text{unsuitable}$$

Figure 4.4: *Equation to determine the accessibility cost of the crossing type of a Way*

4.4.3 Compacting the result

The A* algorithm has found a suitable path from start to finish in the road network, but it contains redundant information because many consecutive hops may share the same Way. So it needs to be simplified before showing it to the user.

The generated path is parsed, and each stretch of consecutive hops that share the same Way ID is replaced by a single hop between the first and last node of the stretch. After this operation, consecutive hops may still share the same road but not the same Way. This is because multiple Ways in the OSM database may represent several stretches of the same road, and each stretch may contain different accessibility features. Therefore, it has been decided not to simplify the directions further.

This simplified path is then encoded to JSON to make the result generic and platform-independent. The JSON will also feature accessibility information for each hop, and other miscellaneous data. The application will then display the result to the end-user, ending the routing process.

Chapter 5

Evaluation

In this chapter, an evaluation of ARPA, the executed project, will be presented. The evaluation consists of two comparisons between ARPA and two commercially available routing solutions: Google Maps and OpenRouteService.

Google Maps has been chosen as it is one of the most popular routing services available at the time of writing. Instead, OpenRouteService has been chosen as it uses OSM as a basis for the road network, and it offers the most accessibility settings between the analysed commercial solutions. When relevant, the two modes of OpenRouteService, introduced in section 3.4, will be distinguished by referring to the Foot API of OpenRouteService as F-OpenRouteService, and by referring to the Wheelchair API of OpenRouteService as W-OpenRouteService. No academic solution will be presented in the evaluation because no publically accessible APIs were available, and no article offered the source code or enough information to recreate their approaches accurately.

The first evaluation will be an analysis of to what extent the three approaches tackle the research question. This is done by comparing the features offered and accessibility requirements taken into account in the routing process.

The second evaluation will analyse the quality of the generated routes of the three approaches. This will be done by introducing three metrics: Reliability,

Accessibility and Quality. Then 50 start and end point pairs will be generated in the city of Heidelberg, Germany. Next, all three approaches will generate two routes between each of the pairs. The first route will be generated with the weights representing a pedestrian with no preferences, and the second route will be generated with the weights representing a wheelchair user. Finally, the three approaches will be compared using the routes they generated applying the introduced metrics.

The chapter will then conclude with a discussion about the results and limits of the evaluation.

5.1 Comparison against research question

In this section, the three approaches will be evaluated against the five goals of the research question. The applications' features will be analysed for each goal to determine to what extent they undertake it.

Goal 1: Generate directions suitable for pedestrians in a low amount of time

Each of the three applications offers directions for pedestrians that maximise the use of pedestrian paths and streets with sidewalks. Furthermore, while "low amount of time" cannot be objectively analysed as internet speed factors into the measurements, all the approaches return a valid route in under 10 seconds, which is considered a good value. Google Maps and OpenRouteService are the fastest, returning the route in a trivial amount of seconds, and ARPA being the slowest with a time closer to 10 seconds. For the ARPA time, it is assumed that the road network has already been downloaded and parsed as the amount of time elapsed in the download operation will vary widely based on the internet connection speed.

An additional consideration has to be made for OpenRouteService. W-OpenRouteService, while still returning a route in a small number of seconds, works only in small areas in a limited amount of cities worldwide, significantly reducing its scope. OpenRouteService does not explicitly disclose these areas as they

depend on the amount of accessibility information available in OpenStreetMap. When sufficient accessibility information is not available, the application will refuse to generate a route.

Goal 2: Take into account the user’s specific needs

When examining the accessibility requirements tackled by the various approaches showcased in table 3.2, there is a clear distinction between Google Maps and the other two approaches. The only feature that Google Maps considers in its routing is the type of road, preferentially routing users to roads suitable to pedestrians, which is also done by the two other approaches. Furthermore, before further examining, there is a need to separate the two versions of OpenRouteService as they offer different accessibility settings.

F-OpenRouteService only allows the option to avoid stairs in the path, while W-OpenRouteService considers all the other requirements listed in the table. While still considering the "Stairs" requirement, ARPA allows the user to further indicate their requirements by allowing to specify if the stairs encountered in the path can be used when accompanied by a handrail, a ramp, or if they present a number of steps lower than a user-defined value.

When considering the "Path Width" and "Slopes" requirements, both ARPA and W-OpenRouteService allow the user to define the minimum width of the sidewalk/road and the maximum incline of the road. Instead, for the "Kerb height" requirement, ARPA only lets users define a preference for low kerbs, while W-OpenRouteService allows the user to define a maximum height.

For the requirement "Surface Type", W-OpenRouteService provides more granular settings than ARPA. While ARPA only lets users set a preference for having smooth roads, W-OpenRouteService lets users define the minimum surface type they can use from a list of surface types ordered by smoothness. Furthermore, it lets users provide the lower bound for the supplemental "smoothness" tag and the track type.

The "Crossing type" requirement is only tackled by ARPA, which lets users choose road crossings that include road markings and traffic signals. Moreover, ARPA allows users to be routed towards roads that feature illumination and roads not shared with cyclists, fulfilling two requirements in the "Walkability" category that are not covered by any other application.

Goal 3: Provide users with accessibility information for every step of the route

Both OpenRouteService and Google Maps only provide minimal information about the route. Both provide the altitude changes in the road. Additionally, OpenRouteService provides information about the surface of the road, along with its type. Instead, ARPA provides all the accessibility information it gathers from each step of the road, highlighting the information relevant to the user.

Goal 4: Allow users to have control of the final generated route

Both OpenRouteService and Google Maps allow the user to obtain up to 3 alternative roads, with unknown criteria for generating them. Instead, ARPA allows users to ban any segment of road and receive another that circumvents said section.

Goal 5: Use publicly available datasets for road and accessibility information

Google Maps is based on privately gathered data with restrictive use for external developers. Both OpenRouteService and ARPA are instead based on OpenStreetMap, which is crowdsourced and free to use.

5.2 Comparison of the general quality of generated routes

This section shows a generalised evaluation of the quality of the generated roads by the three approaches. First, the metrics used for the evaluation will be introduced.

Next, the environment in which the comparison is executed is outlined. Then, how the three applications have been adapted for the evaluation is explained. Lastly, the results of the evaluation are presented and discussed.

5.2.1 Explanation of comparison metrics

The metrics that will be used to evaluate the generated scores are Reliability, Accessibility and Quality. All three metrics are represented with decimal values ranging from 0 to 1, with 1 being the best value. In addition, these metrics are relative to a hypothetical user's preferences defined with the weights described in section 4.2.1. Interesting accessibility information is defined as all the accessibility features with a user-defined weight greater than zero.

Reliability Score

The Reliability Score measures how much accessibility information is available about the generated route. For example, a value of 1 will mean that the generated route contains all the accessibility information interesting to the user. In contrast, a value of 0 means that no interesting accessibility information has been found for the route. This metric is a slightly modified version of the one presented by Neis (38).

The Reliability Score of the generated route will be the sum of the reliability score of all segments multiplied by their length in meters divided by the total length of the route, as seen in Figure 5.3. To calculate this score, each segment will be assigned a sub-reliability score based on what accessibility information the segment has available, only factoring in those interesting to the user, as shown in figures 5.1 and 5.2. Finally, this sub-score is divided by the sum of interesting weights to normalise it between 0 and 1.

$$reliabilityValue_x = \begin{cases} 1, & \text{if Accessibility Information x is available about} \\ & \text{this segment} \\ 0, & \text{if Accessibility Information x is NOT available about} \\ & \text{this segment} \end{cases}$$

Figure 5.1: *Equation to determine the reliability value of a piece of accessibility information*

$$segmentReliability = \frac{\sum_{i=AccessibilityFeature}^{n=AllAccessibilityFeatures} userWeight_i * reliabilityValue_i}{\sum_{i=AccessibilityFeature}^{n=AllAccessibilityFeatures} userWeight_i}$$

Figure 5.2: *Equation to determine the reliability score of a single segment*

$$Reliability\ Score = \sum_{i=Segment\ of\ route}^{n=Generated\ route} segmentReliability_i * \left(\frac{segmentLength_i}{totalLength} \right)$$

Figure 5.3: *Equation to determine the reliability score of a route*

Accessibility Score

The Accessibility Score measures how well the route's available interesting accessibility information aligns with the recommended values. For example, a value of 1 means that the available accessibility information about the generated route indicates that the route is perfectly suitable for the user. In contrast, a value of 0 indicates the opposite.

The Accessibility Score of the generated route will be the sum of the accessibility score of all segments multiplied by their length in meters divided by the total

length of the route, as seen in Figure 5.6. To calculate this score, each segment of the generated route will be assigned a sub-accessibility score based on how close the available accessibility information aligns to the recommended values, only factoring in those interesting to the user, as shown in Figures 5.4 and 5.5. This score will then be divided by the sum of interesting weights of the available information to normalise it between 0 and 1.

$$accessibilityValue_x = \begin{cases} 1, & \text{if Accessibility Information x aligns with the} \\ & \text{recommended value} \\ 0, & \text{if Accessibility Information x does NOT align with} \\ & \text{the recommended value} \end{cases}$$

Figure 5.4: *Equation to determine the accessibility value of a piece of accessibility information*

$$segmentAccessibility = \frac{\sum_{i=AccessibilityFeature}^{n=AvailableAccessibilityFeatures} userWeight_i * accessibilityValue_i}{\sum_{i=AccessibilityFeature}^{n=AvailableAccessibilityFeatures} userWeight_i}$$

Figure 5.5: *Equation to determine the accessibility score of a single segment*

$$Accessibility\ Score = \sum_{i=Segment\ of\ route}^{n=Generated\ route} segmentAccessibility_i * \left(\frac{segmentLength_i}{totalLength} \right)$$

Figure 5.6: *Equation to determine the accessibility score of a route*

Total Score

The Total Score measures the general quality of the generated route. Quality is defined by how much accessibility information is available for the route and how it

aligns with the user's preferences. For example, a value of 1 means that the route contains all the interesting accessibility information and perfectly aligns with the recommended values. In contrast, a value of 0 means that there is no accessibility information or, the generated route is not suitable for the user, or both.

To calculate the Total Score (as seen in Figure 5.7), the Reliability and Accessibility Scores are multiplied to generate a single score.

$$TotalScore = ReliabilityScore * AccessibilityScore$$

Figure 5.7: *Equation to determine the total score of a route*

5.2.2 Comparison environments

The three applications will generate routes from pairs of randomly generated start/end coordinates for the evaluation. These coordinates will be generated inside a bounding box of 3 square kilometres in the city of Heidelberg, Germany. This city has been chosen as it is one of the cities with the most complete information in OSM according to the research discussed in section 2.2.3, and because W-OpenRouteService covers it entirely. The distance between the start and end coordinates will be 1 kilometre as the crow flies. 1 Kilometer has been chosen as the distance because it is a realistic length for a pedestrian journey. Furthermore, while the straight-line distance between the two points is 1 kilometre, the length of the generated routes may vary based on the availability of a path in the road network.

First comparison: No preferences

For the first comparison, 50 pairs of coordinates will be generated. Then, the routers will be tasked with creating a route suitable for a pedestrian for each of the pairs, with the only requirement being a preference for pedestrian paths and sidewalks. This requirement will be encoded as follows for the three applications:

- For Google Maps, its API has been queried with the mode 'walking'

- For OpenRouteService, the Foot version of the API has been queried with no additional parameters
- For ARPA, the router has been queried with all of the user’s weights set to 0

Second comparison: Routing for a wheelchair user

For the second comparison, 50 pairs of coordinates will be generated. Then, the routers will be tasked with creating a route suitable for a wheelchair user for each pair of coordinates. These requirements have been defined using the default settings of the Wheelchair API of OpenRouteService. These requirements will be encoded as follows for the three applications:

- For Google Maps, its API has been queried with the mode ‘walking’
- For OpenRouteService, the Wheelchair version of the API has been queried with the options outlined in table 5.1
- For ARPA, the router has been queried the weights outlined in table 5.2

Option Name	Value
maximum_incline	"6"
maximum_sloped_kerb	"0.06"
minimum_width	1
smoothness_type	"good"
surface_type	"cobblestone"
track_type	"grade1"
avoid_features	["steps"]

Table 5.1: *Parameters used in the API call to the Wheelchair Open Route Service API for the wheelchair generalised evaluation*

5.2.3 Adapting applications for evaluation

The OpenStreetMap ID of the Ways that represent the roads used by the generated routes is required to calculate the metrics utilised in the evaluation. The Way

userWeights	
Weight	Value
sharedBicycles	0
lit	0
surface	0.75
width	0.5
incline	0.5
stairs	1
kerb	0.25
crossing	0
userSettings	
Setting	Value
minWidth	1
maxIncline	6

Table 5.2: *Weights used for ARPA for the wheelchair generalised evaluation*

IDs are necessary to retrieve the accessibility information, which is only present in OSM. Because commercially available APIs might use different databases than OSM and because their service is providing directions, they do not usually offer OSM IDs along with their directions. Instead, they give the users geographic coordinates, road names and textual instructions for following the directions.

This is the case for both Google Maps and F-RouteService, and therefore these coordinated will need to be translated into the equivalent OSM Ways to allow for a standardised comparison. An explanation of how the results of the various APIs have been standardised for the evaluation is provided in the upcoming paragraphs. Furthermore, the potential errors this process has introduced into the final results will be discussed.

Google Maps

The main problem with translating Google Maps coordinates is that it does not use the OSM database. Because of this, there is not any link between the two databases, so an alternative way needs to be introduced to determine equivalent OSM Ways from the coordinates. When providing directions, Google maps pro-

vides a set of steps to reach the destination, and for each step, start and end coordinates are given along with the name of the road.

A geocoder can be used to determine the equivalent road with this information. The most apparent solution, searching the road name, will not work as a road name does not provide enough detail to identify it uniquely. Instead, geocoders can be used in a "reverse" mode, describing the road from its coordinates. As the start and end coordinates are likely to be intersections, which could identify two or more roads, a different coordinate needs to be located. Therefore, the average point between the start and end coordinates is determined, which is likely to be contained in the road. A geocoder is used to reverse search this average point, which will return a road with its Way ID. Doing this operation for all the steps in the directions provided by Google Maps will return a collection of Ways that will be utilised in the evaluation.

There are two main problems with this approach. The first problem is that it does not work when the average coordinate point is not on the road, such as roads that are not straight. The second problem is linked with how Ways are represented in OSM. Because a series of Ways can represent a single road, the result of the geocoding request could map to a Way that represents only a section of the road instead of the entire length.

Manual analysis of a random set of translated roads has been done to determine the impact of these problems. For the first one, the city chosen for the evaluation presents primarily straight roads. For the second one, the problem becomes evident when a significant stretch of the route features the same road. However, it is contrasted by the fact that a road rarely offers different accessibility features on different sections.

F-OpenRouteService

While OpenRouteService uses OSM as a database, like Google Maps, the foot version only provides a set of steps to reach the destination, with start and end co-

ordinates given along with the road's name for each step. However, because it uses OSM as the base for the road graph, the coordinates provided match the coordinates of Nodes stored in OSM. Therefore, by searching the Nodes associated with each pair of coordinates, the path generated can be recreated with OSM Elements.

To do so, first, all the coordinates are translated into the equivalent Nodes. Then, if two subsequent nodes are referenced by the same Way, that Way is the OSM representation of that step, so it is saved. This process is then repeated for all Nodes to generate the equivalent directions with OSM Elements.

The main problem with the translation is that not all coordinates map to an equivalent Node. This happens because, for optimisation purposes, the graph used for the routing presents dynamic nodes calculated after the OSM data is parsed.

Manual analysis of a random set of translated roads has been done to determine the impact of this problem. The analysis has shown that around 80% of the nodes are correctly translated. However, the generated route presents missing information mainly at the start and end of the route, as some optimisation happens with the OpenRouteService Geocoder. Furthermore, dynamic nodes are also present when the directions route a user through open areas such as plazas, which are infrequent, presenting minor consequences.

W-OpenRouteService

Unlike the Foot API of OpenRouteService, the Wheelchair version also allows requesting the OSM Ways used to generate the directions. Therefore, the translation process only involves suitably encoding the data for the evaluation, which does not introduce any possible errors.

Threats to validity

The problems shown with Google Maps and F-OpenRouteService introduce an amount of error in the results. While an objective measurement of the impact cannot be carried out, random manual inspections of the generated routes have

been performed to reduce the error as much as possible. Nevertheless, it has been decided to still introduce and discuss the evaluation results as the errors are deemed small enough to be still relevant. Nevertheless, the measurements of the routes generated by ARPA and W-OpenRouteService are accurate.

5.2.4 Results and discussion

The evaluations executed as specified in section 5.2.2 produced the following results. In the discussion, Google Maps will be used as the baseline.

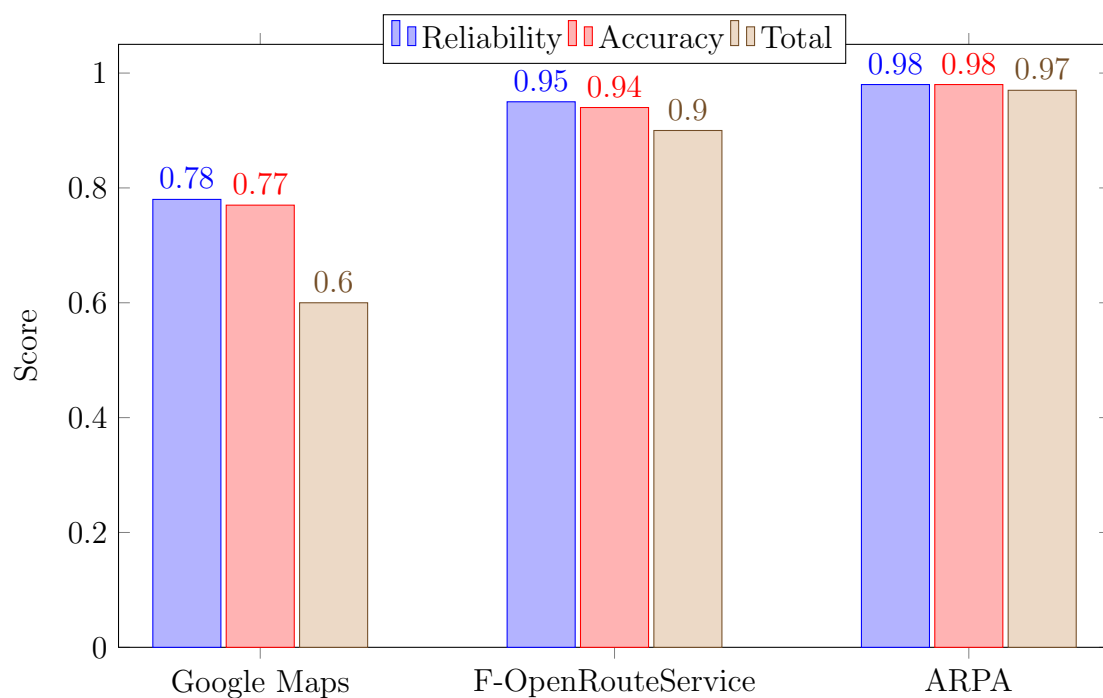


Figure 5.8: Graph showing the results for the generalised evaluation without requirements

Observing the first evaluation results in Table 5.8, it is evident that scores are adequate for all three routers. This is to be expected as the only requirement was routing the users on roads safe for pedestrians. Google Maps, acting as the baseline, guides the user on roads for which there is not always always accessibility information on OSM or on roads that are not suitable for pedestrians. Instead, OpenRouteService and ARPA provide much higher quality directions, achieving

near-perfect accuracy and reliability scores, with ARPA slightly better overall.

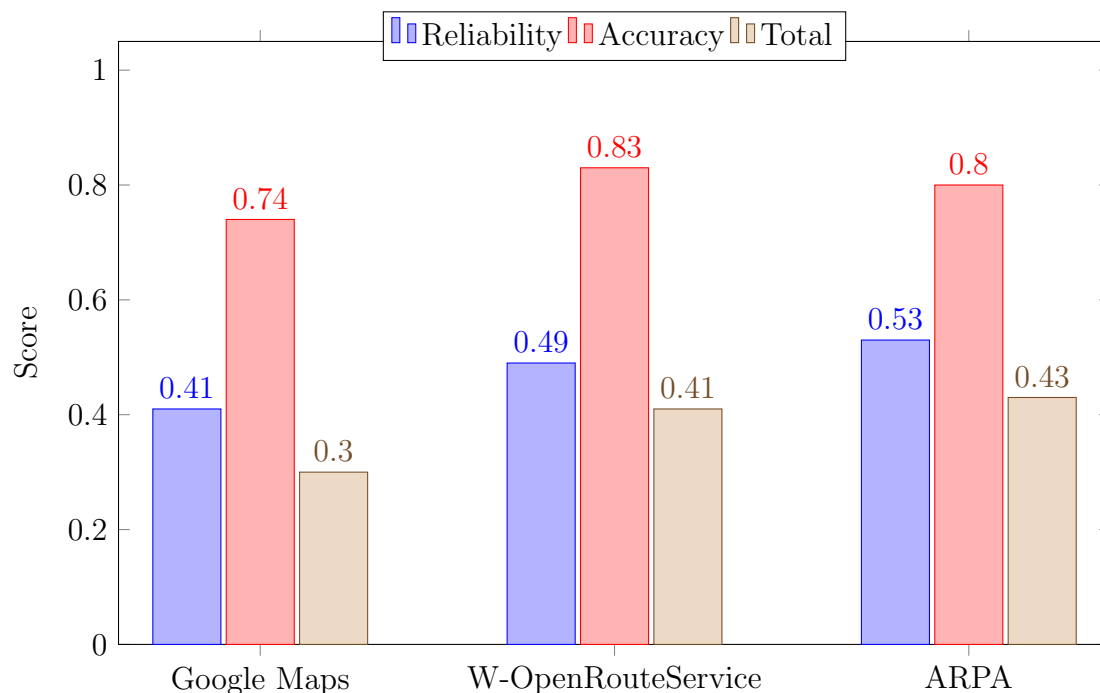


Figure 5.9: Graph showing the results for the generalised evaluation with the requirements set for a wheelchair user

Observing the evaluation results in Table 5.9, the scores are overall worse for the three applications as they were tasked with factoring in six different requirements to represent a wheelchair user. All three approaches present a low reliability score, associated with the lack of full coverage of accessibility information for the road network in the OSM database. An interesting result is the score of Google Maps, as even with the higher number of requirements, it still provides roads similar to the specialised routers. This can be credited to accessibility-focused city planning. While similar, the specialised routers still offer significantly better quality routes. A fascinating insight that can be gleaned from the results is the difference in reliability and accuracy scores between W-OpenRouteService and ARPA. The scores imply that ARPA prefers routing users toward roads containing more accessibility information, even if they encounter slightly less favourable conditions. This difference makes ARPA score marginally better than W-OpenRouteService overall.

5.3 Discussion

The extent to which ARPA fulfils the dissertation’s goals can be gleaned when looking at the first comparison result. Examining the three applications, ARPA is the only one that provides users with accessibility information for every step of the route and allows them to have complete control of the generated route. Furthermore, only ARPA and OpenRouteService use a publicly available dataset for the routing, and both also take into account the most requirements for the routing of a walking impaired person. When analysing the requirements taken into account, ARPA provides the most. Furthermore, ARPA has more granular settings for stairs, while OpenRouteService provides granular settings for suitable surface materials. The defying aspect that makes ARPA better than OpenRouteService is its global availability, as OpenRouteService provides customisable routes for walking impaired persons only in few small areas. At the same time, ARPA manages to deal with the incomplete dataset to still deliver potential routes.

Moreover, the second comparison results also show how ARPA has a slight advantage over the other two approaches. In particular, the results shown in table 5.9 indicate how ARPA generates better quality routes than OpenRouteService when the equivalent requirements are considered even in the rare areas in which the Wheelchair API of OpenRouteService allows the generation of directions.

The combination of both comparisons shows how ARPA stands out and improves against the commercial state of the art of routing for pedestrians and walking impaired persons.

Chapter 6

Conclusion

In this dissertation, ARPA, an Accessibility-Focused Route Planning Assistant, was presented as a tool that provides pedestrian-focused directions to users while considering their specific requirements in generating the routes. The application presented fulfils the requirements outlined in the research question to a great extent.

The directions are generated in under 10 seconds and route the users towards safe pedestrian footpaths. The accessibility requirements taken into account while routing are the most comprehensive among the current state of the art. Furthermore, users are shown all the accessibility information for every step of the route, and they have complete control over the final directions as they can veto any specific segment of the road network and get provided directions that avoid it. Moreover, all the data used in the routing originates from OpenStreetMap, a public and crowdsourced geographic database.

While there are many requirements taken into account when routing, they are not complete. For example, when comparing the requirements outlined in section 2.1.4 with those considered by ARPA, road features like accessible facilities and street furniture are missing. Furthermore, ARPA does not allow users to set a preference for routes with low noise levels and traffic and has no support for temporary accessibility disruptions such as construction work.

Using OSM as the database for accessibility also presents limitations. Many cities do not have enough accessibility information for the road network, limiting the application's usefulness. Furthermore, the lax approach to tagging standards in OSM means that any region needs to be manually analysed to understand its tagging dialect to make sure the routes generated recognise all the information available. For the scope of this dissertation, this has only been done for select regions in Italy, Ireland and Germany.

Future Work

There are three key areas to explore to improve on the work presented by this dissertation, with the first two improving on the requirements outlined by the research question and the last expanding the scope further.

The first area is tackling more accessibility requirements. Accessibility facilities are present in OSM, but they are not linked with any other Element by default, making them difficult to attribute to a specific road. Furthermore, crowdsourcing capabilities can be integrated into ARPA to allow users to report noise and traffic levels, along with street furniture and temporary disruptions.

The second area is refining the routing. When routing users through open spaces, ARPA uses the perimeters to give directions. This can be improved by creating dynamic networks of nodes to represent the open spaces and then route users using this generated information. Furthermore, the time it takes for ARPA to generate a route can be drastically improved by hosting local versions of the external APIs used to download the map and translate the start and end locations, as those are the most time-consuming operations. If necessary, the computational time can be reduced further by rewriting the application using more optimised programming languages such as C++.

The most significant limitation of ARPA is due to the lack of widespread accessibility information in OSM, and as such, a third area that could be explored is

the incentivisation of users to contribute to the OSM project. For example, ARPA could make available to OSM volunteers a list of the most searched roads that are still missing accessibility information so that the most trafficked routes will get the highest priority when mapping.

Bibliography

- [1] T. J. Nechyba and R. P. Walsh, “Urban sprawl,” *Journal of economic perspectives*, vol. 18, no. 4, pp. 177–200, 2004.
- [2] T. Litman and F. Laube, “Automobile dependency and economic development,” *Victoria Transport Policy Institute, Canada*, 2002.
- [3] P. D. Norton, “Fighting traffic: the dawn of the motor age in the american city,” 2011.
- [4] H. Frumkin, “Urban sprawl and public health,” *Public health reports*, 2016.
- [5] J. O. Hill, H. R. Wyatt, G. W. Reed, and J. C. Peters, “Obesity and the environment: where do we go from here?,” *Science*, vol. 299, no. 5608, pp. 853–855, 2003.
- [6] A. Power, “Social exclusion and urban sprawl: is the rescue of cities possible?,” *Regional studies*, vol. 35, no. 8, pp. 731–742, 2001.
- [7] P. A. Jargowsky, “Sprawl, concentration of poverty, and urban inequality,” *Urban sprawl: Causes, consequences, and policy responses*, pp. 39–72, 2002.
- [8] M. J. Alcoforado and H. Andrade, “Global warming and the urban heat island,” pp. 249–262, 2008.
- [9] O. Edenhofer, “Climate change 2014: mitigation of climate change,” vol. 3, 2015.
- [10] S. Gössling, “Why cities need to take road space from cars-and how this could be done,” *Journal of Urban Design*, vol. 25, no. 4, pp. 443–448, 2020.

- [11] S. Gössling, “Integrating e-scooters in urban transportation: Problems, policies, and the prospect of system change,” *Transportation Research Part D: Transport and Environment*, vol. 79, p. 102230, 2020.
- [12] M. Weng, N. Ding, J. Li, X. Jin, H. Xiao, Z. He, and S. Su, “The 15-minute walkable neighborhoods: Measurement, social inequalities and implications for building healthy communities in urban china,” *Journal of Transport & Health*, vol. 13, pp. 259–273, 2019.
- [13] P. Newman and J. Kenworthy, “‘peak car use’: understanding the demise of automobile dependence,” *World Transport Policy & Practice*, vol. 17, no. 2, pp. 31–42, 2011.
- [14] M. A. Alfonzo, “To walk or not to walk? the hierarchy of walking needs,” *Environment and Behavior*, vol. 37, pp. 808–836, 11 2005.
- [15] R. H. Lo, “Walkability: what is it?,” *Journal of Urbanism: International Research on Placemaking and Urban Sustainability*, vol. 2, no. 2, pp. 145–166, 2009.
- [16] R. Ewing and S. Handy, “Measuring the unmeasurable: Urban design qualities related to walkability,” *Journal of Urban Design*, vol. 14, no. 1, pp. 65–84, 2009.
- [17] T. A. Litman, “Economic value of walkability,” *Transportation Research Record*, vol. 1828, no. 1, pp. 3–11, 2003.
- [18] J. D. Marshall, M. Brauer, and L. D. Frank, “Healthy neighborhoods: Walkability and air pollution,” *Environmental Health Perspectives*, vol. 117, no. 11, pp. 1752–1759, 2009.
- [19] J. Speck, *Walkable city: How downtown can save America, one step at a time*. Macmillan, 2013.
- [20] M. H. Matthews, “Private worlds and public places: mapping the environmental values of wheelchair users,” 1995.
- [21] W. H. Organization, “World report on disability,” 2011.

- [22] D. o. E. United Nations and S. Affairs, “Transforming our world: the 2030 agenda for sustainable development,” 2015.
- [23] A. N. Evcil, “Raising awareness about accessibility,” *Procedia-Social and Behavioral Sciences*, vol. 47, pp. 490–494, 2012.
- [24] L. Ferrari, M. Berlingiero, F. Calabrese, and J. Reades, “Improving the accessibility of urban transportation networks for people with disabilities,” *Transportation Research Part C: Emerging Technologies*, vol. 45, 01 2013.
- [25] D. J. Weiss, A. Nelson, H. S. Gibson, W. Temperley, S. Peedell, A. Lieber, M. Hancher, E. Poyart, S. Belchior, N. Fullman, B. Mappin, U. Dalrymple, J. Rozier, T. C. D. Lucas, R. E. Howes, L. S. Tusting, S. Y. Kang, E. Cameron, D. Bisanzio, K. E. Battle, S. Bhatt, and P. W. Gething, “A global map of travel time to cities to assess inequalities in accessibility in 2015,” *Nature*, vol. 553, pp. 333–336, Jan 2018.
- [26] C. Sundling, B. Berglund, M. E. Nilsson, R. Emaradson, and L. R. Pendrill, “Overall accessibility to traveling by rail for the elderly with and without functional limitations: The whole-trip perspective,” *International Journal of Environmental Research and Public Health*, vol. 11, pp. 12938–12968, 12 2014.
- [27] T. Poldma, D. Labbé, S. Bertin, Ève De Grosbois, M. Barile, K. Mazurik, M. Desjardins, H. Herbane, and G. Artis, “Understanding people’s needs in a commercial public space: About accessibility and lived experience in social settings,” *Alter*, vol. 8, no. 3, pp. 206–216, 2014. *Le centre commercial : un laboratoire vivant / Mall as Living Lab* (Editeurs : Michel Desjardins et Isabelle Ville).
- [28] B. Gleeson, “Can technology overcome the disabling city?,” 1999.
- [29] S. Goldsmith, *Universal design*. Routledge, 2007.
- [30] R. Imrie and M. Kumar, “Focusing on disability and access in the built environment,” *Disability & Society*, vol. 13, no. 3, pp. 357–374, 1998.

- [31] R. Kitchin and R. Law, “The socio-spatial construction of (in) accessible public toilets,” *Urban studies*, vol. 38, no. 2, pp. 287–298, 2001.
- [32] T. Völkel and G. Weber, “A new approach for pedestrian navigation for mobility impaired users based on multimodal annotation of geographical data,” in *Universal Access in Human-Computer Interaction. Ambient Interaction* (C. Stephanidis, ed.), (Berlin, Heidelberg), pp. 575–584, Springer Berlin Heidelberg, 2007.
- [33] A. A. Shahraki, “Urban planning for physically disabled people’s needs with case studies,” *Spatial Information Research*, 2020.
- [34] T. Völkel, R. Kühn, and G. Weber, “Mobility impaired pedestrians are not cars: Requirements,” pp. 1085–1093, 2008.
- [35] D. I. für Normung, *Standard specification by the German Institute for Standardization*. 1998.
- [36] C. R. D. United States Department of Justice, “Americans with Disabilities Act, Standards for Accessible Design.” https://www.ada.gov/2010ADASTandards_index.htm, 2010. [Online; accessed June-2021].
- [37] H. Matthews, L. Beale, P. Picton, and D. Briggs, “Modelling access with gis in urban systems (magus): Capturing the experiences of wheelchair users,” *Source: Area*, vol. 35, pp. 34–45, 2003.
- [38] P. Neis, “Measuring the reliability of wheelchair user route planning based on volunteered geographic information,” *Transactions in GIS*, vol. 19, pp. 188–201, 4 2015.
- [39] A. D. Sobek and H. J. Miller, “U-access: A web-based system for routing pedestrians of differing abilities,” *Journal of Geographical Systems*, vol. 8, pp. 269–287, 2006.
- [40] M. Hashemi and H. A. Karimi, “Collaborative personalized multi-criteria wayfinding for wheelchair users in outdoors,” *Transactions in GIS*, vol. 21, pp. 782–795, 8 2017.

- [41] S. Resnikoff, V. C. Lansingh, L. Washburn, W. Felch, T.-M. Gauthier, H. R. Taylor, K. Eckert, D. Parke, and P. Wiedemann, “Estimated number of ophthalmologists worldwide (international council of ophthalmology update): will we meet the needs?,” *British Journal of Ophthalmology*, vol. 104, no. 4, pp. 588–592, 2020.
- [42] S. H. Norgate, “Accessibility of urban spaces for visually impaired pedestrians,” *Proceedings of the Institution of Civil Engineers - Municipal Engineer*, vol. 165, no. 4, pp. 231–237, 2012.
- [43] B. F. G. Katz, S. Kammoun, G. Parseihian, O. Gutierrez, A. Brilhault, M. Auvray, P. Truillet, M. Denis, S. Thorpe, and C. Jouffrais, “Navig: augmented reality guidance system for the visually impaired,” *Virtual Reality*, vol. 16, pp. 253–269, Nov 2012.
- [44] A. S. Helal, S. E. Moore, and B. Ramachandran, “Drishti: An integrated navigation system for visually impaired and disabled,” p. 149, IEEE Computer Society, 2001.
- [45] S. Hennig, F. Zobl, and W. W. Wasserburger, “Accessible web maps for visually impaired users: Recommendations and example solutions,” *Cartographic Perspectives*, vol. 2017, pp. 6–27, 2017.
- [46] W. Heuten, N. Henze, and S. Boll, “Interactive exploration of city maps with auditory torches,” in *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, (New York, NY, USA), p. 1959–1964, Association for Computing Machinery, 2007.
- [47] K. Moustakas, G. Nikolakis, K. Kostopoulos, D. Tzovaras, and M. G. Strintzis, “Haptic rendering of visual data for the visually impaired,” *IEEE MultiMedia*, vol. 14, no. 1, pp. 62–72, 2007.
- [48] Z. Wang, B. Li, T. Hedgpeth, and T. Haven, “Instant tactile-audio map: Enabling access to digital maps for people with visual impairment,” in *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '09, (New York, NY, USA), p. 43–50, Association for Computing Machinery, 2009.

- [49] L. Zeng and G. Weber, “Audio-haptic browser for a geographical information system,” in *Computers Helping People with Special Needs* (K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, eds.), (Berlin, Heidelberg), pp. 466–473, Springer Berlin Heidelberg, 2010.
- [50] S. E. Stock, D. K. Davies, M. L. Wehmeyer, and Y. Lachapelle, “Emerging new practices in technology to support independent community access for people with intellectual and cognitive disabilities,” *NeuroRehabilitation*, vol. 28, no. 3, pp. 261–269, 2011.
- [51] S. Fickas, R. Lemoncello, and M. M. Sohlberg, “Requirements engineering in a mobile setting: How travelers with a cognitive impairment ask for and use help,” in *User Modeling and Adaptation for Daily Routines*, pp. 177–193, Springer, 2013.
- [52] J. Gomez, G. Montoro, J. C. Torrado, and A. Plaza, “An adapted wayfinding system for pedestrians with cognitive disabilities,” *Mobile Information Systems*, vol. 2015, p. 520572, Oct 2015.
- [53] D. D. McMahon, C. C. Smith, D. F. Cihak, R. Wright, and M. M. Gibbons, “Effects of digital navigation aids on adults with intellectual disabilities: Comparison of paper map, google maps, and augmented reality,” *Journal of Special Education Technology*, vol. 30, no. 3, pp. 157–165, 2015.
- [54] E. Gordon, “Epithets and attitudes,” 1983.
- [55] L. M. G. DUHANEY and D. C. DUHANEY, “Assistive technology: Meeting the needs of learners with disabilities,” *International Journal of Instructional Media*, vol. 27, p. 393, 2021/7/16/ 2000. 4.
- [56] S. Alper and S. Raharinirina, “Assistive technology for individuals with disabilities: A review and synthesis of the literature,” *Journal of Special Education Technology*, vol. 21, no. 2, pp. 47–64, 2006.
- [57] J. Howe *et al.*, “The rise of crowdsourcing,” *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

- [58] E. Estellés-Arolas and F. G.-L. de Guevara, “Towards an integrated crowdsourcing definition,” *Journal of Information Science*, vol. 38, no. 2, pp. 189–200, 2012.
- [59] M. F. Goodchild, “Citizens as sensors: the world of volunteered geography,” *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [60] L. See, P. Mooney, G. Foody, L. Bastin, A. Comber, J. Estima, S. Fritz, N. Kerle, B. Jiang, M. Laakso, H. Y. Liu, G. Milènski, M. Nikšieč, M. Painho, A. Podör, A. M. R. Olteanu-Raimond, and M. Rutzinger, “Crowdsourcing, citizen science or volunteered geographic information? the current state of crowdsourced geographic information,” *ISPRS International Journal of Geo-Information*, vol. 5, 5 2016.
- [61] F. E. A. Horita, L. C. Degrossi, L. F. G. de Assis, A. Zipf, and J. P. de Albuquerque, “The use of volunteered geographic information (vgi) and crowdsourcing in disaster management: a systematic literature review,” 2013.
- [62] Y. Sun, H. Fan, M. Bakillah, and A. Zipf, “Road-based travel recommendation using geo-tagged images,” *Computers, Environment and Urban Systems*, vol. 53, pp. 110–122, 2015.
- [63] Y. Sun, A. Mobasheri, X. Hu, and W. Wang, “Investigating impacts of environmental factors on the cycling behavior of bicycle-sharing users,” *Sustainability*, vol. 9, no. 6, p. 1060, 2017.
- [64] A. Mobasheri, J. Deister, and H. Dieterich, “Wheelmap: the wheelchair accessibility crowdsourcing platform,” *Open Geospatial Data, Software and Standards*, vol. 2, p. 27, Nov 2017.
- [65] OpenStreetMap, “Why are you making openstreetmap?.” https://wiki.openstreetmap.org/wiki/FAQ#Why_OpenStreetMap.3F. [Online; accessed July-2021].
- [66] OpenStreetMap, “About openstreetmap.” https://wiki.openstreetmap.org/wiki/About_OpenStreetMap. [Online; accessed July-2021].

- [67] OpenStreetMap, “History of openstreetmap.” https://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap. [Online; accessed July-2021].
- [68] J. Anderson, D. Sarkar, and L. Palen, “Corporate editors in the evolving landscape of openstreetmap,” *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, 2019.
- [69] OpenStreetMap, “Osm.xml.” https://wiki.openstreetmap.org/wiki/OSM_XML. [Online; accessed July-2021].
- [70] OpenStreetMap, “Node.” <https://wiki.openstreetmap.org/wiki/Node>. [Online; accessed July-2021].
- [71] OpenStreetMap, “Way.” <https://wiki.openstreetmap.org/wiki/Way>. [Online; accessed July-2021].
- [72] OpenStreetMap, “Relation.” <https://wiki.openstreetmap.org/wiki/Relation>. [Online; accessed July-2021].
- [73] OpenStreetMap, “Tags.” <https://wiki.openstreetmap.org/wiki/Tags>. [Online; accessed July-2021].
- [74] W. Medjroubi, U. P. Müller, M. Scharf, C. Matke, and D. Kleinhans, “Open data in power grid modelling: new approaches towards transparent grid models,” *Energy Reports*, vol. 3, pp. 14–21, 2017.
- [75] S. Geisler and C. Quix, “Evaluation of real-time traffic applications based on data stream mining,” in *Data Mining for Geoinformatics*, pp. 83–103, Springer, 2014.
- [76] C. Prandi, S. Ferretti, S. Mirri, and P. Salomoni, “Trustworthiness in crowd-sensed and sourced georeferenced data,” pp. 402–407, Institute of Electrical and Electronics Engineers Inc., 6 2015.
- [77] M. F. Goodchild and L. Li, “Assuring the quality of volunteered geographic information,” *Spatial statistics*, vol. 1, pp. 110–120, 2012.

- [78] H. Senaratne, A. Mobasheri, A. L. Ali, C. Capineri, and M. Haklay, “A review of volunteered geographic information quality assessment methods,” *International Journal of Geographical Information Science*, vol. 31, no. 1, pp. 139–167, 2017.
- [79] M. Haklay and C. Ellul, “Completeness in volunteered geographical information—the evolution of openstreetmap coverage in england (2008–2009),” *Journal of Spatial Information Science*, 2010.
- [80] J. Jokar Arsanjani, P. Mooney, A. Zipf, and A. Schauss, *Quality Assessment of the Contributed Land Use Information from OpenStreetMap Versus Authoritative Datasets*, pp. 37–58. Cham: Springer International Publishing, 2015.
- [81] A. Mobasheri, Y. Sun, L. Loos, and A. L. Ali, “Are crowdsourced datasets suitable for specialized routing services? case study of openstreetmap for routing of people with limited mobility,” *Sustainability (Switzerland)*, vol. 9, 2017.
- [82] OpenStreetMap, “Sidewalks.” <https://wiki.openstreetmap.org/wiki/Sidewalks>. [Online; accessed July-2021].
- [83] M. Kada, V. Walter, M. Kada, and H. Chen, “Shortest path analyses in raster maps for pedestrian navigation in location based systems,” 2006.
- [84] H. Holone, G. Misund, and H. Holmstedt, “Users are doing it for themselves: Pedestrian navigation with user generated content,” pp. 91–99, 2007.
- [85] T. Völkel and G. Weber, “Routecheckr: Personalized multicriteria routing for mobility impaired pedestrians,” pp. 185–192, Association for Computing Machinery, 2008.
- [86] C. E. Palazzi, L. Teodori, and M. Rocchetti, “Path 2.0: A participatory system for the generation of accessible routes,” pp. 1707–1711, 2010.
- [87] W. Sasaki and Y. Takama, “Walking route recommender system considering saw criteria,” pp. 246–251, 2013.

- [88] S. Mirri, C. Prandi, P. Salomoni, F. Callegati, and A. Campi, “On combining crowdsourcing, sensing and open data for an accessible smart city,” pp. 294–299, Institute of Electrical and Electronics Engineers Inc., 12 2014.
- [89] D. Quercia, R. Schifanella, and L. M. Aiello, “The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city,” pp. 116–125, Association for Computing Machinery, 2014.
- [90] H. A. Karimi, L. Zhang, and J. G. Benner, “Personalized accessibility map (pam): a novel assisted wayfinding approach for people with disabilities,” *Annals of GIS*, vol. 20, pp. 99–108, 2014.
- [91] A. Mancini and P. Zingaretti, “Point to point navigation for people with mobility impairments,” Institute of Electrical and Electronics Engineers Inc., 10 2014.
- [92] K. A. Kulakov, A. I. Shabaev, and I. M. Shabalina, “The route planning services approach for people with disability,” vol. 2015-June, pp. 89–95, IEEE Computer Society, 6 2015.
- [93] M. S. Rahaman, Y. Mei, M. Hamilton, and F. D. Salim, “Capra: A contour-based accessible path routing algorithm,” *Information Sciences*, vol. 385-386, pp. 157–173, 4 2017.
- [94] A. Gharebaghi, M.-A. Mostafavi, G. Edwards, and P. Fougeyrollas, “User-specific route planning for people with motor disabilities: A fuzzy approach,” *ISPRS International Journal of Geo-Information*, vol. 10, p. 65, 2 2021.
- [95] Route4U. <https://route4u.org/>. [Online; accessed June-2021].
- [96] A. M. Bishop, “Routino.” <http://www.routino.org/>. [Online; accessed June-2021].
- [97] O. R. Service. <https://openrouteservice.org/>. [Online; accessed June-2021].
- [98] G. Maps. <https://www.google.com/maps/>. [Online; accessed June-2021].

- [99] O. R. Service, “What is the algorithm used by ors api for qgis.” <https://ask.openrouteservice.org/t/what-is-the-algorithm-used-by-ors-api-for-qgis/2476/3>, 2021. [Online; accessed June-2021].
- [100] R. Bauer, D. Dellinger, P. Sanders, D. Schieferdecker, D. Schultes, and D. Wagner, “Combining hierarchical and goal-directed speed-up techniques for dijkstra’s algorithm,” *ACM J. Exp. Algorithmics*, vol. 15, Mar. 2010.
- [101] P. Neis and D. Zielstra, “Generation of a tailored routing network for disabled people based on collaboratively collected geodata,” *Applied Geography*, vol. 47, pp. 70–77, 2 2014.
- [102] Python. <https://www.python.org/>. [Online; accessed July-2021].
- [103] Flask. <https://www.python.org/>. [Online; accessed July-2021].
- [104] Jinja. <https://jinja.palletsprojects.com/en/3.0.x/>. [Online; accessed July-2021].
- [105] Nominatim. <https://nominatim.openstreetmap.org/ui/about.html>. [Online; accessed July-2021].
- [106] R. A. Geocoder. <https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding/>. [Online; accessed July-2021].
- [107] O. API. <http://overpass-api.de/>. [Online; accessed July-2021].
- [108] OpenStreetMap, “Tag:highway=pedestrian.” <https://wiki.openstreetmap.org/wiki/Tag:highway%3Dpedestrian>. [Online; accessed July-2021].
- [109] S. Hahmann, J. Miksch, B. Resch, J. Lauer, and A. Zipf, “Routing through open spaces—a performance comparison of algorithms,” *Geo-Spatial Information Science*, vol. 21, pp. 247–256, 7 2018.
- [110] OpenStreetMap, “Key:highway.” <https://wiki.openstreetmap.org/wiki/Key:highway>. [Online; accessed July-2021].

- [111] OpenStreetMap, “Key:kerb.” <https://wiki.openstreetmap.org/wiki/Key:kerb>. [Online; accessed July-2021].
- [112] OpenStreetMap, “Key:crossing.” <https://wiki.openstreetmap.org/wiki/Key:crossing>. [Online; accessed July-2021].
- [113] OpenStreetMap, “Key:width.” <https://wiki.openstreetmap.org/wiki/Key:width>. [Online; accessed July-2021].
- [114] OpenStreetMap, “Key:incline.” <https://wiki.openstreetmap.org/wiki/Key:incline>. [Online; accessed July-2021].
- [115] OpenStreetMap, “Bicycle.” <https://wiki.openstreetmap.org/wiki/Bicycle>. [Online; accessed July-2021].
- [116] OpenStreetMap, “Key:lit.” <https://wiki.openstreetmap.org/wiki/Key:lit>. [Online; accessed July-2021].
- [117] OpenStreetMap, “Key:surface.” <https://wiki.openstreetmap.org/wiki/Key:surface>. [Online; accessed July-2021].
- [118] OpenStreetMap, “highway=steps.” <https://wiki.openstreetmap.org/wiki/Tag:highway%3Dsteps>. [Online; accessed July-2021].

Appendix A

Screenshots of the application

The screenshot displays the ARPA application interface. On the left, a map of Heidelberg shows a green route starting from Heidelberg Zoo and ending at Pfaffengrunder Terrasse. The map includes labels for various locations like SRH Hochschule Heidelberg, Botanischer Garten der Universität Heidelberg, and the Neckar river. On the right, a table provides detailed attributes for the route segments.

Reset Colors		Type of segment			Road Name		
Ban	Highlight	Road			Tiergartenstraße		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Yes	Unknown	Yes	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Hofmeisterweg		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Yes	No	Yes	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Hofmeisterweg		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Yes	Unknown	Yes	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Hofmeisterweg		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Unknown	Unknown	Yes	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Unknown Road Name		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Unknown	Unknown	Unknown	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Unknown Road Name		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Yes	Unknown	Unknown	asphalt	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Unknown Road Name		
Sidewalk	Has cyclists	Lit	Surface Type	Width	Incline	Kerb	
Yes	Yes	Yes	paving_stones	Unknown	Unknown	Unknown	
Ban	Highlight	Road			Im Neuenheimer Feld		

ARPA: Accessibility-focused Route Planning Assitant

Start Goal

Figure A.1: Screenshot of the main page of the application

ARPA: Accessibility-focused Route Planning Assitant

How important is avoiding routes shared with bicycles?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

How important is going on lit roads?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

How important is going on a road with a good surface material?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

How important are short kerbs?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

How important are supervised crossings?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

How important is the width of a road?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

What is the minimum acceptable road width?

How important is the incline of a road?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

What is the maximum acceptable road incline?

How important is avoiding stairs?

Not important	A little important	Important	Very important	Essential	0.00
---------------	--------------------	-----------	----------------	-----------	------

Use stairs if they have a handrail :

Use stairs if they have a ramp :

Use stairs if they have less than this amount of steps :

[Save](#)

Figure A.2: Screenshot of the weights page of the application

Appendix B

Constants and values used in the routing function

Constant	Value
Heuristic multiplier	1.25
Change road cost	0.05
Unknown cost	0.2
Unsuitable cost	0.4
Impossible cost	10

Table B.1: *A table outlining the constants used in the routing function*

OSM Value	Decimal Value
excellent	1
good	0.9
intermediate	0.7
bad	0.4
very_bad	0.2
horrible	0
very_horrible	0
impassable	0

Table B.2: *A table outlining the decimal value equivalent of the surface smoothness*

OSM Value	Decimal Value
grade1	1
grade2	0.6
grade3	0.4
grade4	0.2
grade5	0

Table B.3: *A table outlining the decimal value equivalent of the track types*

OSM Value	Decimal Value
flush	1
lowered	0.9
raised	0

Table B.4: *A table outlining the decimal value equivalent of the kerbs*

OSM Value	Decimal Value
supervised	1
unsupervised	0.8
unmarked	0

Table B.5: *A table outlining the decimal value equivalent of the crossing types*

OSM Value	Decimal Value
paved	1
concrete	1
metal	1
paving_stones	1
asphalt	0.95
chipseal	0.95
wood	0.8
compacted	0.8
fine_gravel	0.6
stone	0.4
sett	0.4
cobblestone	0.3
gravel	0.3
pebblestone	0.3
ground	0.3
dirt	0.3
earth	0.3
unhewn_cobblestone	0.2
stepping_stones	0.2
grass	0.2
unpaved	0.2
rock	0.2
salt	0.2
grass_paver	0.2
mud	0.1
woodchips	0.1
snow	0.1
ice	0.1

Table B.6: *A table outlining the decimal value equivalent of the surface materials*