

**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Fakultät Informatik Institut für Angewandte Informatik

Professur Mensch-Computer Interaktion

Diplomarbeit

Navigationssystem für blinde Fußgänger und ÖPNV-Nutzer

bearbeitet von

Eric Scheibler

geboren am 24.08.1985 in Görlitz

Betreuer: Dipl.-Inf. Martin Spindler
Hochschullehrer: Prof. Dr. rer. nat. habil. Gerhard Weber

Eingereicht am 04.02.2014

Inhaltsverzeichnis

1. Einleitung	1
2. Aktueller Stand von Forschung und Technik	3
2.1. Frühe Entwicklungen	3
2.2. Mobile Endgeräte	4
2.3. Weitere Navigationshilfsmittel	5
2.4. Laufende Forschungsprojekte	6
2.5. Zusammenfassung	7
3. Anforderungsanalyse	8
3.1. Nutzeranforderungen	8
3.2. Akquise des Kartenmaterials	10
3.2.1. Google Maps	10
3.2.2. Navteq und TomTom	11
3.2.3. OpenStreetMap (OSM)	11
3.2.4. Fazit	12
4. Entwurf	13
4.1. Aufbau und Inhalt der Route	13
4.2. Routing-Service	15
4.2.1. Online Routing	15
4.2.2. Eigener Routing Server	18
4.2.3. Routenberechnung auf dem Client	18
4.2.4. Fazit	19
4.3. Client	19
4.4. System-Entwurf	19
4.5. Routingprozess	20
4.5.1. Wahl von Start- und Zielpunkt	20
4.5.2. Fußgängeroute	21
4.5.3. ÖPNV-Route	29
4.5.4. Darstellung der Route auf dem Client	36
4.6. Kreuzungsquerungen	42
5. Implementierung	46
5.1. Server	46
5.1.1. Installation und Konfiguration	46
5.1.2. Aufbau der Komponenten	49

Inhaltsverzeichnis

5.2. Client	51
6. Evaluation	55
6.1. Funktionstest	55
6.2. Nutzertest	59
6.3. ÖPNV Bewertungsfunktion	62
6.4. Zusammenfassung	64
7. Ausblick	65
A. Anhang	67
A.1. Ausschnitt einer Fußgänger-Route	67
A.2. Abgelaufene Fußgängerrouen	72
A.3. Fragebogen für den Nutzertest	76
A.4. Ergebnis der Evaluation der ÖPNV-Routen	78
Literaturverzeichnis	I

Abbildungsverzeichnis

4.1. Darstellung einer Route im GPX Format	17
4.2. Bahnhofsvorplatz des Dresdner Hauptbahnhofs	23
4.3. Kreuzung der Nöthnitzer Straße mit einer Anliegerstraße	29
4.4. Bus- und Straßenbahnhaltestelle Dresden, Cämmerswalder Straße	35
4.5. Screenshot des RoutingClients, Routenübersicht	38
4.6. Screenshot des RoutingClients, Routenverfolgung	39
4.7. Auffinden der Kreuzung durch Differenzwinkelberechnung	41
4.8. Querungsstelle: Modell von Wendor	43
4.9. Querungsmöglichkeit an der Kreuzung Nöthnitzer Straße, Regensburger Straße	44
A.1. Funktionstest Route 1	72
A.2. Funktionstest Route 2	73
A.3. Funktionstest Route 3	74
A.4. Funktionstest Route 4	75

Tabellenverzeichnis

3.1. Zusammenfassung der Vor- und Nachteile der Kartendiensteanbieter . . .	12
4.1. Zusammenfassung der Vor- und Nachteile der Routing Services	19
4.2. Tabelle zur Bestimmung der Routing Gewichte	25
5.1. Ausschnitt aus der Intersections Tabelle	48
5.2. Ausschnitt aus der IntersectionData Tabelle	48
5.3. Ausschnitt aus der erstellten Routengraphen Tabelle	49
6.1. Ergebnis der ÖPNV Verbindungsevaluation	64

1. Einleitung

Mobilität stellt einen wesentlichen Faktor zur Teilhabe am sozialen und beruflichen Alltag dar. Einige Beispiele hierfür sind die Bewältigung des täglichen Arbeitsweges und das Wahrnehmen von Freizeitaktivitäten. Mobilität wird definiert als Fähigkeit einer Person, selbständig und gegebenenfalls durch die Verwendung von Hilfsmitteln einen Wechsel des räumlichen Orts zu vollziehen[14, Abschnitt 1.1].

Um einen entfernten Ort zu erreichen muss eine sogenannte Navigationsaufgabe bewältigt werden. Diese gliedert sich in Mikro- und Makronavigation. Die Mikronavigation bildet den lokalen Anteil der Navigationsaufgabe und dient der Orientierung und Vermeidung von Hindernissen. Die Makronavigation umfasst hingegen den Routenplanungsprozess[10].

Diese Arbeit konzentriert sich auf die Gruppe der blinden und sehbehinderten Nutzer. Diese Nutzergruppe erfährt bei der Lösung der Navigationsaufgabe substanzielle Einschränkungen, da Landmarken und Umgebungsmerkmale nur eingeschränkt oder gar nicht wahrgenommen werden können. Daraus ergibt sich eine hohe mentale Belastung beim Erlernen neuer Wege und der Orientierung in unbekanntem Gebieten.

Diese Einschränkungen verursachen außerdem Probleme bei der Nutzung von Wegbeschreibungen und Navigationssystemen. Einerseits sind die präsentierten Navigationsanweisungen für Blinde oftmals ungenügend. Details wie der Aufbau von Kreuzungen oder die Beschaffenheit von Wegen werden nicht erläutert, da sie vom Nutzer bereits visuell wahrgenommen und somit als redundant erachtet werden. Akustische und taktile Landmarken kommen ebenfalls nicht zum Einsatz. Andererseits gibt es Probleme bei der Zugänglichkeit der Benutzungsoberflächen.

Dabei wäre ein geeignetes Hilfsmittel zur Bewältigung der Navigationsaufgabe gerade für die Gruppe der Sehgeschädigten vorteilhaft, um die mentale Belastung signifikant zu senken und Unsicherheiten in unvertrauten Umgebungen abzubauen. Dadurch könnte eine größere Anzahl von Betroffenen ermutigt werden, unbekannte Orte ohne fremde Hilfe aufzusuchen. Dies würde zur Steigerung der Selbstständigkeit beitragen und die Teilhabe in sozialen und kulturellen Bereichen fördern.

Daher erfolgt im Rahmen dieser Arbeit der Entwurf eines Prototypen, welcher den Nutzer bei der Makronavigation unterstützt und auf die erhöhten Anforderungen der Zielgruppe eingeht. Die Teilziele lassen sich wie folgt beschreiben:

1. Bereitstellung einer Route mit hohem Detailgrad. Dazu gehört unter anderem die ausführliche Beschreibung von Kreuzungen und Wegabschnitten. Blindenspezifische Attribute wie taktile Leitlinien und akustische Ampelanlagen werden ebenfalls berücksichtigt.
2. Einbeziehung des öffentlichen Personennahverkehrs in die Routenberechnung

1. Einleitung

3. Präsentation der Route auf einem barrierefreien Endgerät

Als Erstes wird ein Überblick über den aktuellen Forschungsstand gegeben. Anschließend folgt die Analyse der Nutzer- und Funktionsanforderungen. In den darauffolgenden Kapiteln werden Entwurf und Implementierung des routenbasierten Navigationssystems beschrieben. Dabei liegt der Fokus hauptsächlich auf der Umsetzung der besonderen Anforderungen blinder und sehbehinderter Nutzer an ein solches System. Zusätzliche Aufmerksamkeit gilt ferner der Einbeziehung des öffentlichen Personennahverkehrs in den Routenplanungsprozess. Abschließend wird eine Evaluation sowohl der berechneten Route als auch der Benutzeroberfläche des Prototypen durchgeführt.

2. Aktueller Stand von Forschung und Technik

Anfang der neunziger Jahre begann die Entwicklung von Navigationssystemen für den Consumerbereich. Das Erste kam 1989 unter dem Namen Magellan NAV 1000 auf den Markt. Zunächst waren diese Systeme fast ausschließlich für die motorisierte Fortbewegung konzipiert. Die Ortung erfolgte lediglich auf etwa 100 Meter genau, da das US Verteidigungsministerium 1990 eine künstliche Verschlechterung des GPS Signals anordnete¹. Am 02.05.2000 gab die US Regierung diese Störungspraxis auf, woraufhin sich die Genauigkeit um den Faktor 10 verbesserte². Seitdem sind viele Produkte für andere Zielgruppen wie Fußgänger oder Radfahrer entstanden. Sie richten sich allerdings vorwiegend an den Mainstream. Davon abweichend existiert eine Vielzahl an Nutzergruppen mit mehr oder weniger identischen Anforderungen. So benötigen beispielsweise Rollstuhlfahrer die Möglichkeit, Treppen und Kopfsteinpflaster umgehen zu können. Blinde wiederum sind u.a. auf eine besonders genaue Beschreibung der vor ihnen liegenden Route angewiesen.

Nachfolgend werden einige der existierenden Systeme für blinde Fußgänger näher betrachtet.

2.1. Frühe Entwicklungen

Das erste Forschungsprojekt, welches sich speziell mit der Programmierung eines Navigationssystems für Sehbehinderte befasste, entstand Anfang der 90er Jahre in Großbritannien unter dem Namen MOBIC: A SYSTEM FOR FACILITATING INDEPENDENT MOBILITY AND NAVIGATION FOR BLIND PEOPLE[3]. Die Hardware bestand aus einem tragbaren PC inkl. Tastatur, einem GPS Empfänger und Kopfhörer. Das zu Grunde liegende Kartenmaterial wurde von der Firma Ordnance Survey bereitgestellt. Zunächst wurde zu Hause eine Route unter Eingabe des Start- und Zielpunktes berechnet. Diese konnte anschließend virtuell abgeschritten werden um sich mit der zurückzulegenden Strecke bereits im Vorhinein vertraut zu machen. Beim nachfolgenden Ablaufen der Route wurde der Nutzer durch passende Routinganweisungen unterstützt sowie auf umliegende Points of Interest (POI) aufmerksam gemacht. Bei der abschließenden Evaluation schätzten alle Probanden das System als sehr hilfreich ein. Dies galt insbesondere für die Erkundung unbekannter Gebiete. Trotz dieser positiven Rückmeldungen gaben

¹<http://www.techhive.com/article/2000276/a-brief-history-of-gps.html> abgerufen am 29.12.2013

²<http://www.gps.gov/systems/gps/modernization/sa> abgerufen am 22.11.2013

die Forscher zu bedenken, dass einer kommerziellen Vermarktung insbesondere die hohen Lizenzgebühren für das Kartenmaterial entgegenstünden.

1997 brachte die Firma Arkenstone ein auf diesen technischen Grundlagen basierendes Navigationssystem namens Strider auf den Markt³. Neben GPS Empfänger und Sprachsynthesizer verfügte das mehrere Tausend Dollar teure Produkt über das gesamte Kartenmaterial der USA. Auf Grund der bereits erwähnten Störung des GPS Signals erfolgte die Positionsbestimmung allerdings lediglich auf 100 Meter genau.

Im Jahr 2003 begann die Firma Humanware mit dem Verkauf eines der ersten wirklich portablen Navigationssysteme für Blinde. Der Trekker bestand aus einem PDA, welcher mittels Sprachausgabe und taktilem Interface an die Bedürfnisse blinder Nutzer angepasst wurde. Er beinhaltete für Fußgänger angepasstes Kartenmaterial. Unterwegs konnte sich der Nutzer u.a. Straßennamen, Kreuzungen und POI's ansagen lassen. Inzwischen wurde ein Nachfolgegerät mit ähnlichem Funktionsumfang namens Trekker Breeze entwickelt⁴. Nachteilig wirken sich die recht ungenaue Beschreibung der Kreuzungspunkte sowie der recht hohe Anschaffungspreis aus.

2.2. Mobile Endgeräte

In den folgenden Jahren wurde die Entwicklung der mobilen Endgeräte und deren Betriebssysteme stetig vorangetrieben. Dies betraf auch die Zugänglichkeit für Blinde und Sehbehinderte. Der Vorteil bestand darin, dass die Gerätehardware nicht mehr aufwendig selbst entwickelt oder an die Bedürfnisse der blinden Nutzer angepasst werden musste.

Die drei von Blinden am häufigsten genutzten mobilen Betriebssysteme sind:⁵

1. iOS (Apple)
2. Android (Google)
3. Symbian (Nokia)

Symbian wurde von den Firmen Ericsson, Motorola, Nokia und Psion eingesetzt und weiterentwickelt. Seit 2008 besitzt Nokia sämtliche Rechte. Es läuft auf einer Vielzahl an Mobilgeräten u.a. von Nokia und Samsung. Damit es von Blinden verwendet werden kann, muss allerdings ein kostenpflichtiger Screenreader installiert werden. Ein Beispiel hierfür ist Talks⁶ der Firma Nuance Communications. Die Unterstützung und Weiterentwicklung durch Nokia wurde Ende 2012 komplett eingestellt.

³<http://www.senderogroup.com/products/gps/allgps.htm> abgerufen am 29.12.2013

⁴http://www.humanware.com/en-europe/products/blindness/talking_gps/trekker_breeze/_details/id_101/trekker_breeze_handheld_talking_gps.html abgerufen am 20.08.2013

⁵<http://webaim.org/projects/screenreadersurvey4/#os> abgerufen am 01.02.2014

⁶<http://www.nuance.com/for-individuals/mobile-applications/talks-zooms/index.htm> abgerufen am 30.12.2013

2. Aktueller Stand von Forschung und Technik

Apples Betriebssystem iOS läuft seit 2007 ausschließlich auf Apple eigener Hardware wie dem iPhone. Mit dem Update auf die Version 3 wurde das Betriebssystem vom Hersteller selbst um einen Screenreader namens VoiceOver erweitert⁷.

Ähnlich verhält es sich mit Google's Betriebssystem namens Android, welches 2008 veröffentlicht wurde. Google liefert ebenfalls seit 2009 den Screenreader Talkback⁸ mit.

Sowohl VoiceOver als auch Talkback sind kostenlos. Beide werden auf Geräten eingesetzt, welche fast ausschließlich auf die Eingabe mittels Touch Display setzen. Dies erforderte die Entwicklung eines neuen Bedienkonzeptes. Wird ein auf dem Bildschirm dargestelltes Objekt berührt, so werdenzunächst dessen Name und Eigenschaften durch eine Sprachausgabe vorgelesen. Erst ein Doppelklick auf das betreffende Objekt führt dessen eigentliche Aktion aus.

Im dritten Quartal 2013 lag der Marktanteil von Android bei 81,0%, gefolgt von iOS mit 12,9%⁹. Symbian wird nicht mehr separat aufgeführt. Damit sind die beiden mobilen Betriebssysteme mit dem höchsten Marktanteil für Blinde und Sehbehinderte zugänglich.

2.3. Weitere Navigationshilfsmittel

Mit Wayfinder Access¹⁰ stand ein weiteres Navigationssystem für Blinde bereit. Neben der Routenberechnung bot es den Zugriff auf eine POI Datenbank. Mit der Übernahme von Wayfinder Systems durch Vodafone wurde der Service eingestellt.

Neben den extra für Sehbehinderte programmierten Anwendungen sind auch eine Reihe von Mainstream Apps zugänglich. Als Beispiel sei auf Navigon¹¹ für das iPhone verwiesen. Navigon bietet ebenfalls ein Routing an und dessen Programmoberfläche ist mit dem Screenreader VoiceOver gut bedienbar. Von Nachteil ist, dass die Routenführung nicht an die Bedürfnisse Blinder angepasst und damit zu ungenau ist. Des Weiteren verfolgen Programme wie Navigon trotz eines vorhandenen Profils für Fußgänger ein recht straßenzentriertes Routing. Kleinere Fußwege sind oft nicht im Kartenmaterial enthalten und werden somit nicht berücksichtigt.

Neben den klassischen Navigationssystemen, welche dem Nutzer eine Route vom Start zum Zielpunkt errechnen, sind in den letzten Jahren auch verstärkt alternative Hilfsmittel zur Orientierung in unbekanntem Gebieten entstanden.

Eines der Ersten ist das Programm Loadstone¹², welches auf Nokia's Betriebssystem Symbian läuft. Loadstone liefert kein Kartenmaterial mit. Stattdessen legt der Nutzer selbst für ihn wichtige Punkte an. Diese werden danach zu Routen zusammengefasst und können anschließend der Reihe nach abgelaufen werden. Ein Vorteil besteht darin, dass auch Wegpunkte fernab des digitalisierten Kartenmaterials erfasst werden können. Dafür

⁷<http://www.apple.com/de/accessibility/ios/voiceover/> abgerufen am 30.12.2013

⁸<https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback> abgerufen am 30.12.2013

⁹<http://www.idc.com/getdoc.jsp?containerId=prUS24442013> abgerufen am 27.01.2014

¹⁰<https://github.com/wayfinder/?id=912&lang=en-UK> abgerufen am 30.12.2013

¹¹<https://itunes.apple.com/de/app/navigon-europe/id320279293?mt=8> abgerufen am 30.12.2013

¹²<http://www.loadstone-gps.com>, abgerufen am 11.11.2013

ist Loadstone für die Erkundung vollkommen unbekannter Gebiete nicht geeignet, da für das Setzen der Markierungspunkte mindestens einmal eine Hilfsperson benötigt wird.

Ende 2010 wurde ein auf Loadstone aufbauendes Projekt namens LoroDux vorgestellt[5]. Es läuft ebenfalls unter Symbian und wurde in der Programmiersprache J2ME entwickelt. Neben den von Loadstone bekannten Funktionen wurde die Integration von Points of Interest aus der OpenStreetMap (OSM) angestrebt. So soll auch das Erkunden einer unbekanntem Umgebung ermöglicht werden. Ein Routing ist allerdings nicht vorgesehen.

Die iPhone App MyWay¹³ wurde vom Schweizer Blindenverband in Auftrag gegeben. Die Funktionsweise lehnt sich stark an die von Loadstone und LoroDux an. Wie bei diesen Programmen liegt der Fokus auf der Erstellung eigener Routen durch das Setzen von Landmarken. Aber ähnlich wie bei LoroDux ist der Import von zuvor am PC erstellten OSM POI-Listen möglich. So kann beispielsweise eine Datei mit allen Bushaltestellen der Stadt erzeugt werden. Auch hier ist kein Routing von einem beliebigen Start- zum Zielpunkt möglich.

Das Programm Ariadne¹⁴ verfolgt einen etwas anderen Ansatz. Ariadne's Hauptfeature stellt die Erkundung eines Kartenausschnittes dar. Der Nutzer kann den Verlauf von Straßen und Wegen ertasten und ist so in der Lage, sich ein Bild von seiner Umgebung zu machen.

Mit keiner der speziell für Blinde vorhandenen Anwendungen ist ein Routing möglich. Dazu muss auf Programme wie Navigon ausgewichen werden, die allerdings unzureichende Routinganweisungen zur Verfügung stellen.

2.4. Laufende Forschungsprojekte

Neben den praktischen Ansätzen gibt es auch eine Vielzahl an Forschungsprojekten, welche sich dem Problem angenommen haben. Beispielfhaft werden an dieser Stelle zwei noch laufende Projekte vorgestellt.

Das ARGUS Projekt (Assisting peRsonal GUidance System for people with visual impairment) verfolgt einen routenbasierten Ansatz[7]. Der Nutzer wird auf einem sicheren Weg durch städtische und ländliche Gebiete geführt und rechtzeitig vor Hindernissen gewarnt. Die Route besteht aus kontinuierlich zur Verfügung gestellten akustischen und haptischen Anweisungen. Neben der sicheren Passage zum gewünschten Ziel soll es dem Nutzer auch ermöglicht werden, sich eine mentale Karte seiner Umgebung anzulegen.

Derzeit läuft außerdem das Projekt InMoBS¹⁵, an dem u.a. die TU Braunschweig beteiligt ist. InMoBS steht für Innerstädtische Mobilitätsunterstützung für Blinde und Sehbehinderte. Auch hier soll der Nutzer bei der Navigation durch innerstädtische Gebiete unterstützt werden. Die Forschungsschwerpunkte unterscheiden sich jedoch von denen des Argus-Projekts. Ein Schwerpunkt stellt die Erforschung der hochgenauen Personenortung dar. Das europäische Satellitenortungssystem Galileo spielt dabei eine

¹³<https://itunes.apple.com/us/app/myway-classic/id494282724?mt=8> abgerufen am 30.12.2013

¹⁴<https://itunes.apple.com/de/app/ariadne-gps/id441063072> abgerufen am 30.12.2013

¹⁵<http://www.inmobs.de/> abgerufen am 30.12.2013

entscheidende Rolle. Außerdem ist die Vernetzung von Licht-Signal-Anlagen an innerstädtischen Verkehrsknotenpunkten geplant. Der Nutzer soll so sowohl direkt zu den Ampelmasten geroutet werden, wie auch einen besseren Überblick über die Kreuzung erhalten.

2.5. Zusammenfassung

Die vorherigen Abschnitte gaben einen Überblick über den aktuellen Stand der Navigationshilfsmittel für blinde und sehbehinderte Nutzer. Die folgende Aufzählung fasst die Rechercheergebnisse zusammen:

- Die speziell für Blinde und Sehbehinderte entwickelten Lösungen bieten kein oder nur ein sehr eingeschränktes Routing, dienen eher zur Erkundung der Umgebung, sind veraltet oder sehr preisintensiv.
- Die verfügbaren Mainstream-Routing Angebote berücksichtigen wenig bis gar nicht die Bedürfnisse blinder Nutzer.
- Die Forschungsprojekte haben entweder kein flächendeckendes Routing zum Ziel oder befinden sich noch in der Entwicklungsphase. Zur ersten Gruppe werden auch Projekte gezählt, welche die Installation von zusätzlicher Hardware in der Routingumgebung erfordern.
- Der ÖPNV wird bisher kaum in die Streckenplanung einbezogen.

Aufbauend auf diesen Erkenntnissen wird im nächsten Kapitel eine Anforderungsanalyse durchgeführt.

3. Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an ein routenbasiertes Navigationssystem für blinde und sehbehinderte Nutzer zusammengetragen. Nach der Auflistung der Nutzeranforderungen folgen Abschnitte zur Wahl des Kartenmaterials und der Anforderungen an das Clientsystem.

3.1. Nutzeranforderungen

Dieser Abschnitt gibt zunächst einen kurzen Überblick über die von Blinden und Sehbehinderten verwendeten Hilfsmittel und Techniken zur Orientierung und Navigation. Anschließend werden konkrete Anforderungen an ein Navigationssystem aufgelistet.

Ein Navigationssystem ist ein Sekundärhilfsmittel, d.h. es wird in den meisten Fällen als ein zusätzliches Hilfsmittel verwendet. Um es optimal einsetzen zu können, müssen bei den blinden und sehbehinderten Nutzern bereits grundlegende Kenntnisse in Bezug auf die Orientierungsfähigkeiten in bekannten und unbekanntem Gebieten vorhanden sein[13]. Diese Kenntnisse werden zumeist während eines Orientierungs- und Mobilitätstrainings (O&M) vermittelt. Sie umfassen die Orientierung, d.h. die Fähigkeit, die eigene Position zu bestimmen und mit Objekten in der Umgebung in Relation zu setzen. Anschließend folgt das Erlernen von Routen zu häufig aufzusuchenden Orten. Nach Abschluss des Trainings ist der Schüler in der Lage, bekannte Wege ohne fremde Hilfe zu begehen und unbekannte Wege in relativ kurzer Zeit zu erlernen.

Der weiße Langstock stellt ein Primärhilfsmittel dar und wird beinahe von allen Blinden aber auch von einigen Sehbehinderten eingesetzt. Er dient sowohl zur Hinderniserkennung als auch zur Kennzeichnung und ist somit ein geeignetes Hilfsmittel zur Unterstützung der Mikronavigation. Blinde verwenden teils auch einen Blindenführhund zur Vermeidung von Hindernissen oder taktile Karten, um sich einen Überblick über das vor ihnen liegende Gebiet zu verschaffen. Sehbehinderte nutzen hingegen verstärkt Sehhilfen wie beispielsweise Lupen oder Monokulare[14, Abbildung 3.3].

Die folgenden Abschnitte enthalten eine Übersicht über die Anforderungen, welche blinde und sehbehinderte Personen an ein Navigationssystem stellen. Die Daten wurden hauptsächlich aus drei verschiedenen Umfragen zusammengetragen: Der Deutsche Blinden- und Sehbehindertenverband veröffentlichte 2008 die Ergebnisse eines Fragebogens, welcher sich mit den Anforderungen Blinder und Sehbehinderter an ein GPS System befasste[13]. Wenn nicht anders vermerkt, stammen die Anforderungen aus diesem Dokument. Im gleichen Jahr stellte Thorsten Völkel in seiner Doktorarbeit die Resultate einer Umfrage zur Mobilität Sehbehinderter vor[14]. Schließlich wertete das InMoBS-Projekt im Jahr 2013 einen weiteren Fragebogen mit ähnlicher Thematik aus[12].

Zunächst ist festzustellen, dass die Mehrzahl von Blinden und Sehbehinderten überhaupt an Unternehmungen in unbekanntem Umgebungen interessiert ist. Nahezu alle Befragten würden ein auf die Bedürfnisse Sehbehinderter abgestimmtes Navigationssystem einsetzen[14, Abschnitt 3.2.3].

Anforderungen

A1: Allgemeine Anforderungen

A1.1: Das Navigationssystem muss den Nutzer sicher an sein Ziel bringen und sowohl zu Fuß wie auch in öffentlichen Verkehrsmitteln einsetzbar sein.

A1.2: Eine präzise Positionsbestimmung ist unbedingt erforderlich. Als wünschenswert wird eine Abweichung um einen Meter genannt. Liegt kein ausreichend genaues Signal an, so ist dies dem Nutzer mitzuteilen.

A1.3: Die Planung und Simulation der Route muss bereits von zu Hause aus möglich sein.

A1.4: Weiterhin ist der öffentliche Personennahverkehr mit in die Routenplanung einzubeziehen. Dies gilt insbesondere in gut erschlossenen Gebieten, wo der ÖPNV eine wichtige Rolle bei der Beförderung einnimmt. Auf diese Weise kann ein Großteil einer unbekanntem Strecke mit dem Bus oder der Bahn zurückgelegt werden und die Bewältigung einer unbekanntem Route wird dem Nutzer erleichtert. Die Einbeziehung und ggf. sogar Bevorzugung des ÖPNV gilt besonders, wenn sich die Verkehrsbetriebe für die Barrierefreiheit ihres Angebotes einsetzen[1][14, Abschnitt 3.3.4].

So existiert beispielsweise in Prag und Dresden ein Informationssystem für blinde und sehbehinderte Fahrgäste¹. Mit einem kleinen Handsender kann der Außenlautsprecher von Bussen und Bahnen aktiviert und so das Fahrziel erfragt werden. Des Weiteren ist es dem Nutzer möglich, den Fahrer über eine möglicherweise benötigte Hilfestellung zu informieren.

A2: Anforderungen an das verwendete Kartenmaterial

A2.1: Das Kartenmaterial muss einem hohen Detailgrad unterliegen, also beispielsweise auch Wege fernab von Straßen enthalten. Daneben sind Informationen über die Beschaffenheit, Breite und Steigung der Wege, Lage der Fußwege, Treppen sowie Unterführungen zu integrieren[8, Tabelle 1]. Außerdem sind blindenspezifische Attribute wie taktile Leitlinien und akustische Ampeln zu berücksichtigen.

A2.2: Aus dem InMoBS Fragebogen geht ferner hervor, dass die Beschreibung von Kreuzungen von den blinden Teilnehmern als unbedingt notwendig eingestuft wurde. Dies beinhaltet u.a. die Anzahl und Namen der Straßen, den Kreuzungsaufbau bzw. die Kreuzungsart sowie das Vorhandensein von Mittelinseln. Ferner möchte die Mehrzahl der Befragten über namentliche Querstraßen auf der Route informiert werden sowie ihren eigenen Standort markieren[12, Abschnitt 2.2.2].

A2.3: Das Kartenmaterial muss auf dem aktuellen Stand gehalten werden.

A3: Die Zugänglichkeit des Ausgabegerätes muss gewährleistet sein. Dies ist der Fall, wenn die folgenden Unteranforderungen erfüllt werden[2]:

¹<http://www.dvb.de/de/oepnv-fuer-alle/> abgerufen am 31.12.2013

A3.1: Das Gerät darf nicht zu groß und zu schwer sein, um unterwegs bequem in einer Hand gehalten werden zu können. Ferner müssen GPS Empfänger, Kompass und mobiles Internet integriert sein.

A3.2: Das Betriebssystem stellt während der Bedienung ein durchgängiges akustisches Feedback, bestehend aus Sprachausgabe und Hinweistönen bereit. Für Sehbehinderte ist die Anpassung von Kontrast und Schriftgröße vorgesehen. Erfolgt die Steuerung des Klienten durch Gesten auf einem berührungsempfindlichen Bildschirm, so wird bei der Berührung eines dargestellten Objekts dessen Inhalt und Funktion vorgelesen. Anderenfalls wird auf die Eingaben der Tastatur reagiert.

A3.3: Der Aufbau und die Gestaltung der Oberfläche einer erstellten Applikation ist konsistent. Die einzelnen Komponenten wie die Menüleiste und die Buttons zum Bestätigen eines Dialoges befinden sich immer an der gleichen Stelle. So wird der Suchaufwand minimiert. Alle Objekte der grafischen Oberfläche besitzen eine aussagekräftige Beschreibung.

A3.4: Haptische Rückmeldungen durch Vibration unterstützen den Nutzer bei Navigation und kündigen Ereignisse an. Sie sind besonders hilfreich, wenn sich der Nutzer in lauten Umgebungen befindet.

3.2. Akquise des Kartenmaterials

Im folgenden Abschnitt wird ein Überblick über die vorhandenen Kartenanbieter gegeben und deren Eignung für dieses Vorhaben überprüft. Konkret muss deren Datenbestand den Anforderungen aus dem Abschnitt A2 des vorherigen Kapitels genügen.

3.2.1. Google Maps

Google² bietet seit 2005 eine API für ihren Maps Dienst an. Das Kartenmaterial ist insbesondere im Fußgängerbereich recht detailliert. Die Darstellung und Routenberechnung im Browser ist kostenlos möglich. Die erstellte Route ist im Json oder XML Datenformat auch herunterladbar.

Allerdings kann das Routing nur bedingt an die Bedürfnisse des Nutzers angepasst werden. So sind beispielsweise Attribute wie akustische Ampelanlagen und taktile Leitstreifen nicht im Kartenmaterial enthalten. Weitere Details wie der Aufbau von Kreuzungen sind ebenfalls nicht abrufbar. Außerdem darf eine abgerufene Route nicht losgelöst von der zugehörigen Karte, sondern immer nur als Ergänzung verwendet werden³.

²<https://developers.google.com/maps> abgerufen am 27.01.2014

³<https://developers.google.com/maps/documentation/directions/> abgerufen am 27.01.2014

3.2.2. Navteq und TomTom

Navteq⁴ und TomTom⁵ (früher Tele Atlas) sind die zwei größten kommerziellen Kartenanbieter. Beide halten ihre Karten mit speziell ausgerüsteten Fahrzeugen auf dem aktuellen Stand. Diese erfassen neben veränderten Straßenverläufen u.a. auch Geschwindigkeitsbegrenzungen und Einbahnstraßen. In den letzten Jahren wurden vermehrt auch Fußwege mit aufgenommen. Dennoch liegt der Fokus weiterhin auf der Routenerstellung für Fahrzeuge (Verbleich von [19, Figure 13] und [19, Figure 14]). Des Weiteren fehlen vermehrt Points of Interest, die besonders für Fußgänger relevant sind. Darunter fallen beispielsweise Haltestellen des öffentlichen Personennahverkehrs.

3.2.3. OpenStreetMap (OSM)

OpenStreetMap ist ein Projekt zur Erstellung einer freien Weltkarte. Die Daten werden von der Community zusammengetragen, stehen unter der Open Database License (ODbL)⁶ und sind somit für jeden kostenlos abrufbar. OSM wurde von Steve Coast im Juli 2004 als Reaktion auf die bis dahin nur kommerziell verfügbaren Kartenangebote gegründet⁷.

Die Daten liegen in einer Wiki-ähnlichen Datenbank vor. Die drei wichtigsten Strukturen sind:

- Node: Ein einzelner Punkt besteht mindestens aus den Attributen Längen- und Breitengrad. Ferner werden Ersteller und Datum erfasst. Ein Node kann daneben noch viele weitere Tags erhalten. Im Falle eines POI's sind das beispielsweise Name, Typ, ggf. genaue Adresse und Kontaktdaten.
- Way: Ein Weg besteht aus einer Liste von Nodes. So kann ein Weg z.B. einen Straßenabschnitt definieren. Außerdem wird der Typ des Weges (Autobahn, Bundesstraße, Nebenstraße, Fußweg, ect.) und der Name festgehalten.
- Relation: Eine Relation besteht aus Wegen und Knoten. Beispiele hierfür sind Wanderrouen oder der Fahrtverlauf einer bestimmten Straßenbahnlinie.

Im OSM Wiki findet sich neben vielen anderen auch die OSM for the blind Gruppe⁸. Auf dieser Seite finden sich einige Hinweise zur Optimierung der OSM bezüglich der Bedürfnisse von blinden und sehbehinderten Benutzern. Die Genauigkeit der erfassten Positionsdaten steht an oberster Stelle. Weiterhin sollen POI's erfasst werden, welche für Blinde von besonderem Interesse sind. Dazu gehören zum Beispiel Kinos, Theater o.Ä. mit Audiodeskription oder Restaurants mit Braille Karten.

Weitere Attribute sind:

⁴<http://here.com/navteq-redirect/> abgerufen am 27.01.2014

⁵<http://www.tomtom.com> abgerufen am 27.01.2014

⁶<http://opendatacommons.org/licenses/odbl/1-0> abgerufen am 29.01.2014

⁷http://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap#Founding_and_Early_History abgerufen am 27.01.2014

⁸http://wiki.openstreetmap.org/wiki/OSM_for_the_blind abgerufen am 19.08.2013

3. Anforderungsanalyse

- Gebäudeeingänge
- Beschaffenheit des Bodenbelags
- akustische Ampelanlagen
- Taktile Leitlinien an Bahnsteigen oder Kreuzungen
- Sonstige Details wie die Höhe von Bordsteinkanten und die Anzahl von Stufen einer Treppe

Die Abdeckung der Open Street Map ist allerdings recht unterschiedlich. Während Straßen deutschlandweit bis auf wenige Ausnahmen flächendeckend verzeichnet sind, ist dies bei Fußwegen abhängig von der Lage. In städtischen Ballungsgebieten ist ein sehr hoher Detailgrad zu beobachten. In ländlichen Regionen wird dieser oft nicht erreicht. Allerdings wird mittelfristig eine Annäherung der Kartenabdeckung prognostiziert[19].

Ähnlich verhält es sich mit den aufgezählten Tags, welche die Navigation verbessern können. Während Attribute wie die Oberfläche eines Weges noch von vielen Nutzer verwendet und somit recht häufig anzutreffen sind, gibt es bei den blindenspezifischen Tags wie beispielsweise der Verzeichnung von akustischen Ampelanlagen noch große Lücken.

3.2.4. Fazit

Nach der Betrachtung der verschiedenen Kartendienste und der Abwägung der Vor- und Nachteile kommen im Folgenden die Daten der OpenStreetMap zum Einsatz. Die Tabelle 3.1 fasst die Vor- und Nachteile der jeweiligen Kartendiensteanbieter zusammen.

Einerseits weicht die Genauigkeit der OSM Daten nicht signifikant von den Tele Atlas Daten ab [4]. Andererseits sieht die OSM die Erfassung blindenspezifischer Meta-Daten vor, welche in keiner der kommerziellen Datenbanken verzeichnet sind[15, Kapitel 7].

Schließlich spricht auch die freie Lizenz für den Datenbestand aus dem OpenStreetMap Projekt. Zum Einen ist die Nutzung der Karten daher kostenlos. Zum Anderen ist dadurch der Zugriff auf die Rohdaten möglich. Dies ermöglicht einen hohen Grad der Individualisierbarkeit, der von keinem kommerziellen Kartenanbieter geboten wird.

	Google	TomTom	OSM
Kartenmaterial Straßen (A2.1)	hoch	hoch	hoch
Kartenmaterial Fußwege (A2.1)	hoch	niedrig	hoch
Kartenabdeckung (A2.1)	hoch	hoch	mittel
Blindenspezifische Attribute (A2.1)	nein	nein	teilweise
Kreuzungsinformationen (A2.2)	nein	nein	ja
Aktualisierung des Kartenmaterials (A2.3)	kostenlos	kostenpflichtig	kostenlos

Tabelle 3.1.: Zusammenfassung der Vor- und Nachteile der Kartendiensteanbieter

4. Entwurf

Der Systementwurf gliedert sich in die folgenden Unterkapitel. Zuerst werden Aufbau und Inhalt der Route beschrieben. Danach folgt die Festlegung der Projektstruktur. Im nächsten Abschnitt wird der gesamte Routingprozess sequenziell erläutert. Den Abschluss bildet die Vorstellung eines erweiterten Modells zur Darstellung von kreuzungsquerungen.

4.1. Aufbau und Inhalt der Route

Bevor damit begonnen werden kann, die Architektur des Prototypen zu konzipieren, muss zunächst der Aufbau der Routen-Datenstruktur beschrieben werden.

Eine Route besteht zunächst aus einer Reihe von Wegpunkten, die durch Segmente verbunden sind. Anhand der Anforderungen A1.4, A2.1 und A2.2, welche in Kapitel 3.1 aufgestellt wurden, lassen sich vier verschiedene Wegpunkttypen definieren:

- **WayPoint:** Dies ist der Basistyp für alle weiteren Punkttypen. Er besitzt die folgenden Attribute:
 - Name: Der zugrunde liegende Straßenname oder ggf. Straßentyp
 - Geographische Koordinaten (Längen- und Breitengrad)
 - Turn: Alle Wegpunkte mit Ausnahme von Start und Ziel enthalten die Richtungsänderung in Grad. Diese wird aus der Differenz der Richtungen des vorherigen und nächsten Segments errechnet und wird für die Routinganweisungen benötigt. Beim exakten Abbiegen nach rechts läge der Turn beispielsweise bei 90°.
- **POI:** Ein Point of Interest kann eine Adresse oder auch ein Restaurant sein und besteht daher je nach Typ zusätzlich aus Adress- oder Kontaktattributen. Befindet er sich innerhalb eines Gebäudes oder ist selbst ein Gebäude, so werden ferner der Gebäudename sowie dessen Eingänge angegeben.
- **PublicTransport:** Darunter fallen Bus- und Straßenbahnhaltestellen sowie Bahnhöfe. Sie erweitern POI's um Informationen über die haltenden Linien, d.h. Liniennummer, Fahrziel und die nächsten Abfahrten.
- **Kreuzungen:** Für eine genaue Beschreibung einer Kreuzung werden die Koordinaten und Namen jeder mündenden Straße gespeichert. Es existiert also eine Liste von WayPoints, jeweils ergänzt um die Richtung des abgehenden Weges in Grad. Des Weiteren verfügt jede Kreuzung über einen Namen und sofern vorhanden über eine Auflistung der zugehörigen Ampelpositionen.

4. Entwurf

Allerdings werden lediglich Wegpunkte, welche mindestens eines der folgenden Relevanzkriterien erfüllen, in die Route aufgenommen:

- Der Punkt ist eine Kreuzung:
 - an der in eine andere Straße abgelenkt werden muss. Dabei ist nicht unbedingt eine Straßenquerung notwendig, z.B. wenn der Nutzer auf der rechten Straßenseite läuft und nach rechts abbiegen muss.
 - bei welcher eine Straße überquert werden muss, auch wenn die Route beispielsweise geradeaus verläuft. Ein Beispiel hierfür ist eine T-Kreuzung, bei welcher der Nutzer der Hauptstraße folgt.
- Der Punkt ist ein POI oder eine Haltestelle des ÖPNV und wird explizit als Start, Ziel oder Teilziel definiert.
- Einfache WayPoints werden aufgenommen, wenn sich die Richtung (Turn) der Route verändert, also zum Beispiel eine Kurve folgt.

Zwei Wegpunkte werden stets mit einem Wegsegment verbunden. Es existieren zwei Segmenttypen:

- Fußweg: Das Wegsegment wird zu Fuß zurückgelegt. Es besitzt die folgenden Eigenschaften:
 - Name: üblicherweise der Straßename oder falls nicht verfügbar der Straßentyp
 - Länge in Meter
 - Himmelsrichtung in Grad
 - Sofern verfügbar die Oberflächenbeschaffenheit der Straße und Position der Fußwege. Diese Informationen stammen direkt aus den Eigenschaften des Wegsegments. So kann das Tag „sidewalk“ beispielsweise jedem Weg zugewiesen werden. Mögliche Werte sind: no, left, right oder both. Fußwege, die zum Beispiel durch einen Grünstreifen von der Straße getrennt verlaufen, werden derzeit allerdings als eigene Wege erfasst und nicht zur benachbarten Straße gezählt.
 - Eine Liste von POI's entlang des Wegstückes. Alle Punkte mit Ausnahme der einfachen WayPoints, welche nicht den oberhalb definierten Relevanzkriterien entsprechen, werden in diese Liste aufgenommen. So lässt sich während der Navigation ein erweiterter Detailgrad bei der Routenverfolgung einstellen.
- ÖPNV: Das Wegstück wird per Bus oder Bahn zurückgelegt:
 - Liniennummer und Fahrziel
 - Abfahrts- und Ankunftszeit, sowie die Fahrdauer
 - Wenn verfügbar, die Anzahl und Namen der Zwischenhalte

Eine Route ist valide, wenn sie aus einem Start- und Zielpunkt besteht und nie zwei Punkte aufeinanderfolgen, ohne dass ein Segment diese verbindet.

Zur Veranschaulichung des Aufbaus befindet sich der Ausschnitt einer Route im Anhang des Dokuments im Abschnitt A.1. Der Routenausschnitt weist sowohl zu Fuß wie auch mit ÖPNV zurückzulegende Streckenabschnitte auf.

4.2. Routing-Service

Wie im Abschnitt 3.2.4 dargelegt kommt das Kartenmaterial der OpenStreetMap für das Routing zum Einsatz. Im Folgenden wird die Architektur des Routing Service festgelegt. Dabei geht es um die Frage, wo die Route berechnet wird.

Die folgenden Ansätze werden näher betrachtet:

- Die Route wird von einer der verfügbaren Online-basierten OSM Routing Engines berechnet und an den Client übertragen.
- Ein eigens betriebener Routing-Server berechnet die Route und sendet sie an den Client.
- Das Routing findet direkt auf dem Ausgabegerät statt.

Allen Ansätzen ist gemein, dass ein mobiles Endgerät zur Präsentation der Route benötigt wird.

4.2.1. Online Routing

Es existieren mehrere Online Routing Engines, die für einen gegebenen Start- und Zielpunkt eine Route berechnen. Beispiele hierfür sind OpenRouteService¹ und eine Routino Instanz der TU Dresden².

Ein klarer Vorteil liegt im recht geringen Implementierungsaufwand, da die Algorithmen zur Wegfindung und die OSM Datenbank bereits vorhanden sind. Im Open Street Map Wiki findet sich eine Übersicht über vorhandene Online Routing Engines[16]. In dieser wird zur Zeit kein Router aufgeführt der eine, an die Bedürfnisse Blinder angepasste Route berechnet. Des Weiteren wird keine der zugehörigen Webseiten der Online-Router von vornherein als barrierefrei eingestuft.

Nur ein Teil der Anbieter ermöglicht den Export der Route im GPS Exchange Format (GPX) und kommt somit als Routenlieferant in Frage. Das GPX Format ist ein XML-basiertes Datenformat zur Speicherung und zum Austausch von Geo-Daten. Der GPX Export wird benötigt, um die berechnete Route auf den Client zu übertragen und dort darstellen zu können.

Das GPX Format besteht aus den folgenden Datentypen³:

¹<http://openrouteservice.org> abgerufen am 21.08.2013

²<https://routino.arch.tu-dresden.de/routino/router.html> abgerufen am 21.08.2013

³<http://www.topografix.com/gpx/1/1/> abgerufen am 04.01.2014

4. Entwurf

- Der Wegpunkt besteht aus den Geo-Koordinaten (Längen- und Breitengrad) sowie weiteren Attributen wie z.B. Name oder Typ
- Eine Route enthält eine geordnete Liste von Wegpunkten, welche den Weg zu einem Zielpunkt beschreiben.
- Der Track besteht aus einer Liste von Wegpunkten, die einen Linienzug ergeben, z.B. bei der Aufzeichnung eines Pfades durch ein GPS-Gerät.

Zur Veranschaulichung folgt im Quelltextabschnitt 4.1 die Präsentation eines Beispiels. Die dargestellte Route bringt den Nutzer von der Fakultät Informatik der TU Dresden zur Straßenbahnhaltestelle Dresden Münchner Platz. Die Daten stammen von der Webseite OpenRouteService.org⁴.

Auf Grund des standardisierten Formats ist eine Route im GPX Format leicht in die eigene Anwendung zu importieren. Allerdings werden einige wichtige Informationen vom Anbieter nicht zur Verfügung gestellt. Die Anforderung A2.1 aus Kapitel 3.1 verlangt, dass Wege und Straßen mit Namen und weiteren Eigenschaften beschrieben werden. Diese fehlen bei der abgerufenen Route vollständig.

Außerdem sind die Kreuzungsdaten mangelhaft denn über Anzahl und Attribute der abzweigenden Straßen ist so gut wie nichts bekannt. Somit kann auch A2.2 aus Kapitel 3.1 nicht erfüllt werden.

Keine der Online Routing Engines stellt eine Route bereit, welche auf die Bedürfnisse blinder und sehbehinderter Nutzer gesondert eingeht. Weiterhin verletzen die exportierten Routen zwei der gestellten Hauptanforderungen aus Kapitel 3.1. Daher sind die Online Routing Engines zur Bewältigung der aufgabenstellung nicht geeignet.

⁴<http://www.OpenRouteService.org/> abgerufen am 27.01.2014

4. Entwurf

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <gpx
3   xmlns="http://www.topografix.com/GPX/1/1"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://
   www.topografix.com/GPX/1/1/gpx.xsd"
6   creator="www.OpenRouteService.org" version="1.1">
7   <trk>
8     <name>Route_2013.8.21_1536</name>
9     <trkseg>
10      <trkpt lon="13.7227901" lat="51.0257694" />
11      <trkpt lon="13.722815781882744" lat="51.02591772529957" />
12      <trkpt lon="13.722243" lat="51.0260169" />
13      <trkpt lon="13.7222751" lat="51.0261039" />
14      <trkpt lon="13.722224" lat="51.0261112" />
15      <trkpt lon="13.7222136" lat="51.0263778" />
16      <trkpt lon="13.7220545" lat="51.0271266" />
17      <trkpt lon="13.7220057" lat="51.0273357" />
18      <trkpt lon="13.7219965" lat="51.0273799" />
19      <trkpt lon="13.7219202" lat="51.0277079" />
20      <trkpt lon="13.7218836" lat="51.0278653" />
21      <trkpt lon="13.7217802" lat="51.0283098" />
22      <trkpt lon="13.7217344" lat="51.0285101" />
23      <trkpt lon="13.7215546" lat="51.0292969" />
24      <trkpt lon="13.7221256" lat="51.0294826" />
25      <trkpt lon="13.7222488" lat="51.0295216" />
26      <trkpt lon="13.7222996" lat="51.029539" />
27      <trkpt lon="13.7226433" lat="51.0296522" />
28      <trkpt lon="13.722494157471461" lat="51.029824359666414" /
>
29      <trkpt lon="13.7226131" lat="51.0299274" />
30    </trkseg>
31  </trk>
32 </gpx>
```

Abbildung 4.1.: Darstellung einer Route im GPX Format. Startpunkt: Dresden Nöthnitzer Straße 46, Zielpunkt: Straßenbahnhaltestelle Dresden Münchner Platz.

4.2.2. Eigener Routing Server

Eine weitere Möglichkeit ist das Betreiben eines eigenen Routingsservers. Auf diesem wird eine Kopie der OSM Datenbank eingerichtet und mittels eines geeigneten Algorithmus selbst eine Route berechnet.

Ein klarer Vorteil ist die hohe Individualisierbarkeit. Wird die Route komplett eigenständig erstellt, kann in besonderem Maße auf die Nutzeranforderungen A2.1 und A2.2 aus Kapitel 3.1 eingegangen werden. Dies gilt natürlich unter der Bedingung, dass die jeweiligen Informationen in der OpenStreetMap auch verzeichnet sind.

Dafür muss ein Server eingerichtet und dauerhaft betrieben werden. Um eine Route erstellen zu können, wird unter Anderem ein Datenbankserver für Geo-Daten und ein Routingalgorithmus benötigt. Anschließend muss eine Kopie der OpenStreetMap Daten importiert und stetig aktualisiert werden. Zur Lösung dieser Aufgaben existieren allerdings bereits Komponenten, durch deren Verwendung ein Großteil des Implementierungsaufwands eingespart werden kann.

4.2.3. Routenberechnung auf dem Client

Hierbei übernimmt die clientseitige Software sämtliche Schritte von der Routenberechnung bis zur Routenverfolgung durch den Nutzer. Die im Kapitel 2.3 vorgestellte Anwendung Navigon ist dieser Kategorie zuzuordnen.

Größter Vorteil dieser Lösung ist, dass gänzlich auf einen Server verzichtet werden kann und die Bereitschaft des Systems nicht von Dritten abhängt. Die Karten werden einmalig auf das Endgerät geladen. Wird auf einige weitergehende Features wie das Abfragen der aktuellen Abfahrten an einer Haltestelle verzichtet, kommt das Gerät sogar komplett ohne Internetverbindung aus und ist somit auch in Gegenden ohne Mobilfunkabdeckung oder im Ausland einsatzbereit.

Da die Routen ebenfalls selbst erstellt werden, gelten für diesen Bereich die gleichen Vorteile wie beim Betreiben des eigenen Routing Servers.

Nachteilig wirkt sich hingegen die, im Vergleich zu einem Server, geringere Rechenleistung des Clients aus. So muss ggf. mehr Zeit für die Routenberechnung eingeplant oder das Datenmodell vereinfacht werden. Außerdem ist zu bedenken, dass auch nach einer Reduktion des Kartenmaterials eine erhebliche Datenmenge vorverarbeitet und gespeichert werden muss.

Weiterhin sind benötigte Softwarekomponenten zwar für Server aber nicht für mobile Betriebssysteme verfügbar. Darunter fallen beispielsweise die Datenbankerweiterung für Geo-Daten oder Algorithmen zur Aktualisierung des Kartenmaterials und der Routenberechnung. Diese Komponenten sind daher selbst zu entwickeln und erhöhen somit den Implementierungsaufwand drastisch. Dies erschwert wiederum eine potentielle Portierung auf ein weiteres Betriebssystem.

4.2.4. Fazit

Die Online Routing Engines kommen nicht zum Einsatz, da die Anforderungen an die erstellte Route nicht erfüllt werden können. Für die reine Clientlösung spricht, dass weitestgehend auf eine Internetverbindung verzichtet werden kann und keine laufenden Kosten entstehen. Im Gegensatz dazu steigt der Implementierungsaufwand im Vergleich zur serverseitigen Lösung signifikant an, da einige Softwarekomponenten, die für Serverbetriebssysteme bereits verfügbar sind, selbst entwickelt werden müssen. Für die weitere Konzeptionierung wird daher das Betreiben eines eigenen Routingsservers favorisiert. Tabelle 4.1 fasst die Anforderungen nochmals zusammen.

	Online Routing Engine	Eigener Routing Server	Routing nur auf Client
Route erfüllt Anf. A2.1 und A2.2	nein	ja	ja
Aufwand Kartenaktualisierung (A2.3)	keiner	mittel	hoch
Implementierungsaufwand	gering	mittel	hoch
Laufende Kosten	nein	ja	nein
Internetverbindung benötigt	ja	ja	nein

Tabelle 4.1.: Zusammenfassung der Vor- und Nachteile der Routing Services

4.3. Client

Allen Ansätzen ist gemein, dass ein Client für die Darstellung und das Verfolgen der Route benötigt wird. Zur Wahl stehen die von Blinden am Häufigsten eingesetzten mobilen Betriebssysteme: iOS, Android und Symbian (siehe Kapitel 2.2). Symbian wird seit einiger Zeit nicht mehr weiterentwickelt und hat ferner den Nachteil, dass ein Screenreader kostenpflichtig erworben werden muss. Sowohl bei Android als auch iOS gehört er hingegen bereits zur Standardausstattung. Beide Systeme sind für Blinde und Sehbehinderte somit ohne weitere Kosten zugänglich. Sie erfüllen daher die Anforderung A3.2 aus Kapitel 3.1. Android Geräte sind allerdings zu einem wesentlich geringeren Preis erhältlich. Da außerdem die Hard- und Software für die Entwicklung von iOS Apps im Rahmen dieser Diplomarbeit nicht zur Verfügung stand, wird der prototypische Client für das Android Betriebssystem ab Version 4.1 entwickelt. Auf Grund der Client-Server Architektur ist eine spätere Portierung auf iOS relativ einfach durchführbar.

4.4. System-Entwurf

Das Gesamtsystem besteht aus einer Server- und einer Clientkomponente. Zum Server gehören :

4. Entwurf

- Ein Webserver, der die Clientanfragen entgegennimmt und beantwortet.
- Ein Programm, das ausgehend von Start und Ziel eine Route berechnet, ÖPNV Verbindungen und die nächsten Abfahrten einer Haltestelle bereitstellt sowie die POI in der Nähe eines Koordinatenpaares auflistet.
- Eine Datenbank, welche das Kartenmaterial vorhält.

Der Client wählt den Start- und Zielpunkt der Route, übermittelt beides an den Server und erhält nach kurzer Wartezeit die abzulaufende Route zurück. Diese wird anschließend Punkt für Punkt abgelaufen. Unterwegs sind stets Entfernung, Richtung und weitere Informationen zum nächsten Routenpunkt abrufbar.

Dieser skizzierte Ablauf wird im nächsten Abschnitt ausführlich erläutert.

4.5. Routingprozess

Als Nächstes folgt eine ausführliche Beschreibung des Routingprozesses. Dieser umfasst alle Teilschritte von der Routenberechnung bis zum Ablaufen selbiger auf dem Clientsystem.

Ziel ist das Lösen der in Kapitel 1 definierten Navigationsaufgabe unter Berücksichtigung der Anforderungen aus Kapitel 3.1.

Dieser Prozess ist grob in drei Teilaufgaben zu gliedern:

1. Auswahl des Start- und Zielpunktes
2. Routenberechnung: Entweder ausschließlich zu Fuß oder unter Einbeziehung des ÖPNV
3. Verfolgung der errechneten Route auf einem Endgerät

4.5.1. Wahl von Start- und Zielpunkt

Die Wahl des Start- und Zielpunktes wird auf dem Client vorgenommen. Der Nutzer kann zwischen den folgenden Methoden wählen:

1. Eingabe der Start- oder Zieladresse. Dafür müssen mindestens die Straße, Hausnummer und der Ort angegeben werden. Weitere Daten wie PLZ oder Ortsteil sind nicht unbedingt erforderlich. Als Eingabemodalitäten sind sowohl Tastatur als auch Sprache vorgesehen. Die Spracheingabe ist, eine gute Erkennungsrate vorausgesetzt, eine zeitsparende Alternative zur Tastatureingabe. Dies gilt besonders für die virtuellen Tastaturen, wie sie bei Tablets und Smartphones üblich sind. Die Tastatur dient als Fallback für den Fall, dass eine Eingabe per Sprache nicht möglich ist. Dies gilt beispielsweise für laute Umgebungen oder wenn die Privatsphäre gewahrt werden soll. Nach der Eingabe der Adresse werden deren GPS Koordinaten ermittelt.

4. Entwurf

2. Eine weitere Möglichkeit besteht in der Verwendung der aktuellen Position als Startpunkt. Diese Option ist natürlich nur bei einem Client mit einem GPS Empfänger verfügbar. Nachdem die Koordinaten bekannt sind, ermittelt der Client die zugehörige Adresse.
3. Der Nutzer wählt den Start- oder Zielpunkt aus einer Liste von Points of Interest aus. Der Übersichtlichkeit halber werden die POI in verschiedene Kategorien aufgeteilt. Mögliche Kategorien sind beispielsweise:
 - ÖPNV Haltestellen
 - Restaurants, Bars, Kneipen ect.
 - Kreuzungen in der Nähe

Neben der Übersicht über alle POI in der Nähe hat der Nutzer auch die Möglichkeit, nach einem weiter entfernten Punkt zu suchen.

Bei manchen Gebäuden sind die Eingänge separat verzeichnet. Ist dies der Fall, kann der Nutzer den Start- oder Zielpunkt direkt aus der Liste der Gebäudeeingänge wählen. Dies kommt besonders bei großen Gebäuden zum Tragen, wo anderenfalls nur ein ungefährender Punkt in der Gebäudemitte angegeben werden kann.

Die gewählten Start- und Zielpunkte werden intern gespeichert und stehen dem Nutzer zukünftig ebenfalls wieder zur Verfügung. Der Vorteil beim Anlegen einer solchen History ist, dass bereits angesteuerte Punkte nicht erneut gesucht werden müssen.

4.5.2. Fußgängeroute

Dieser Abschnitt erläutert sämtliche Teilschritte zur Berechnung einer Fußgängeroute. Gemäß der gewählten Architektur erfolgt die Berechnung der Route auf einem Server. Auf diesem läuft ein Datenbanksystem und ein von außen erreichbarer Webserver. Der Webserver nimmt die vom Client gesendeten Start- und Zielkoordinaten entgegen und ruft die Funktion zur Erstellung der Route auf.

Bevor dieser Prozess näher beschrieben werden kann, muss zunächst auf den Aufbau der OSM Datenbanken eingegangen werden. Wie bereits im Kapitel 3.2.3 beschrieben, besteht die OpenStreetMap aus Nodes, Ways und Relations. Diese werden jeweils in einer Tabelle der OSM Datenbank vorgehalten. Sie beschreiben sowohl Lage als auch Attribute aller verzeichneten Punkte und Wege. Allerdings lässt sich allein mit diesen Daten noch kein Routing durchführen, da die Verwendung eines Routingalgorithmus einen Graphen voraussetzt. Der Graph muss daher zunächst aus den vorhandenen Informationen gebildet werden. Die Ecken entstammen der Nodes Tabelle und repräsentieren Wegpunkte entlang eines Weges. Sie können beispielsweise eine Kurve oder eine Kreuzung mehrerer Straßen darstellen. Die Kanten des Graphen symbolisieren die eingeschlossenen Wegstücke.

Um den Graphen abzulegen, wird eine neue Tabelle innerhalb der OSM Datenbank mit dem Namen „routing“ erstellt. Jede Zeile dieser Tabelle entspricht einer Kante im Graphen. Um die Kante vollständig zu beschreiben, werden mindestens die folgenden Informationen benötigt:

4. Entwurf

- Startpunkt: Source Id, GPS Koordinaten
- Zielpunkt: Target Id, GPS Koordinaten
- Name und Länge der Kante in Meter
- Wegtyp, Beispiele: Hauptstraße, Nebenstraße, Fußweg ect.

Sind die Voraussetzungen erfüllt, kann die Route berechnet werden. Dazu müssen zunächst Start- und Zielknoten des Graphen ermittelt werden. Anschließend folgt der Aufruf des Routingalgorithmus. Schließlich wird die Route zusammengesetzt.

Start- und Zielknoten des Graphen

Der Client übermittelt für Start und Ziel der Route lediglich die GPS Koordinaten. Daher muss für die Koordinaten zunächst jeweils der optimale Knoten im Graphen gefunden werden. Um dies zu lösen, sind zwei Ansätze denkbar:

1. Das Auffinden des Knotens, welcher dem Start- oder Zielpunkt am nächsten liegt.
2. Die Berechnung der nächst liegenden Kante

Anhand von Abbildung 4.2 wird die Lösung des Problems beispielhaft veranschaulicht. Der Kartenausschnitt zeigt den Bahnhofsvorplatz des Dresdner Hauptbahnhofs einschließlich der naheliegenden Bus- und Straßenbahnhaltestellen. Der Startpunkt S, an welchem sich der Nutzer befindet, ist die stadteinwärts liegende Straßenbahnhaltestelle der Linie 3. Weg A symbolisiert die nach Norden führende Straßenbahnschiene. Weg B ist ein Fußweg, liegt orthogonal zu Weg A und führt zum Haupteingang des Bahnhofs. Ziel ist das Auffinden des optimalen Startknotens im Routinggraphen.

Wird der erste der gerade vorgeschlagenen Lösungsansätze gewählt, so ergeben sich die folgenden Abstände:

- Startpunkt S → Wegpunkt A1: 22 Meter
- Startpunkt S → Wegpunkt A2: 30 Meter
- Startpunkt S → Wegpunkt B1: 7 Meter

Für den zweiten Ansatz wurden folgende Abstände ermittelt:

- Startpunkt S → Weg A: 0 Meter
- Startpunkt S → Weg B: 7.4 Meter

Bei den Werten handelt es sich jeweils um den geringsten Abstand zwischen Punkt und Gerade. Damit steht die Straßenbahnschiene (Weg A) als naheliegendste Kante im Routinggraphen fest. Um anschließend den Startknoten zu ermitteln, wird jeweils der Abstand vom Startpunkt zu den Anfangs- und Endknoten der Kante (Weg A) errechnet. Wegpunkt A1 ist 22 Meter und Wegpunkt A2 30 Meter entfernt. Da sich ein Punkt

4. Entwurf

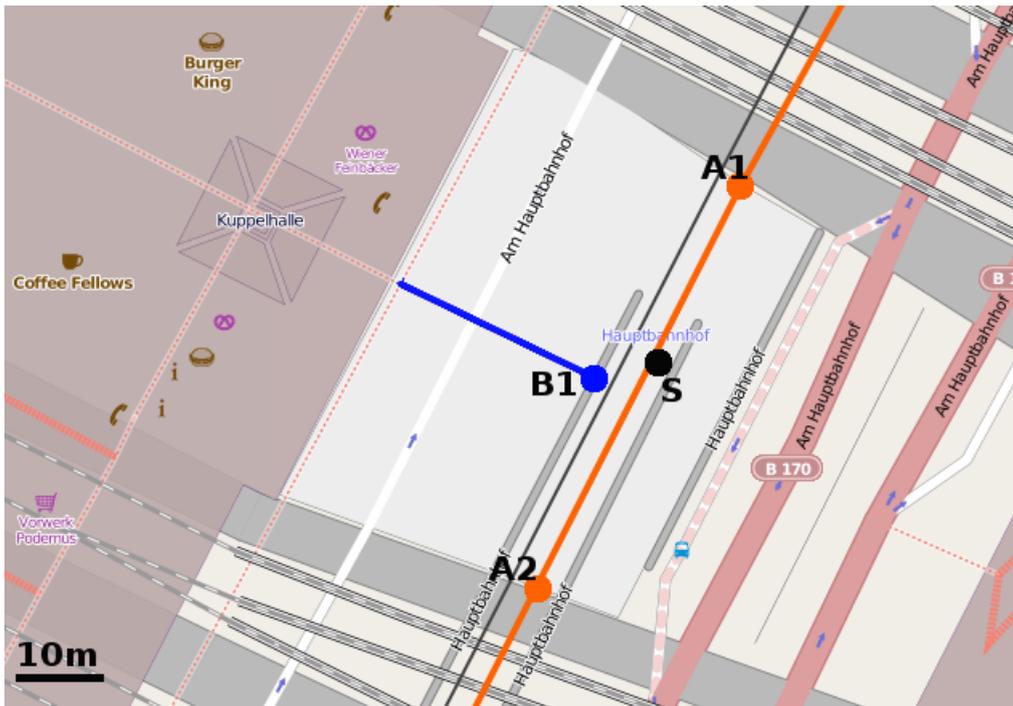


Abbildung 4.2.: Das Bild zeigt den Bahnhofsvorplatz des Dresdner Hauptbahnhofs. Weg A bezeichnet die von Süden nach Norden verlaufende Straßenbahnschiene. Zwei Wegpunkte auf der Schiene sind besonders hervorgehoben: Wegpunkt A1 im Norden und Wegpunkt A2 im Süden. Der Startpunkt ist die östlich von Weg A gelegene Straßenbahnhaltestelle der Linie 3. Westlich der Schiene beginnt an Wegpunkt B1 ein Weg namens Weg B. Dieser verläuft in westlicher Richtung etwa auf den Bahnhofseingang zu. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

auf einer Straßenbahnschiene allerdings nicht als Startpunkt für eine Fußgängeroute eignet, muss schlussendlich noch überprüft werden, ob der anvisierte Wegpunkt Teil einer Kreuzung mit einem Fußweg ist. Bei Wegpunkt A2 ist dies nicht der Fall, A1 hingegen ist sowohl Teil der Straßenbahnschiene als auch eines Fußweges, welcher die Schiene orthogonal kreuzt. Der letzte Schritt ist allerdings nur nötig, wenn es sich um einen für Fußgänger unzugänglichen Wegtyp wie Schienen oder Wasserstraßen handelt.

Werden die Ergebnisse der beiden Lösungsansätze evaluiert, wird klar, dass Wegpunkt A1 dem Knoten Wegpunkt B1 als Start- oder Zielknoten vorzuziehen ist und das, obwohl Wegpunkt B1 drei mal näher an der Haltestelle liegt als Wegpunkt A1. Angenommen, der Eingang des Hauptbahnhofs wäre das Routenziel und B1 der gewählte Startknoten so würde die Überquerung der Straßenbahnschienen nicht in den Routinganweisungen auftauchen. Bei der Wahl von A1 hingegen würde genau mit der Überquerung der Straßenbahnschienen begonnen werden.

4. Entwurf

Die beste Strategie zum Auffinden der Start- und Zielknoten ist daher die Suche nach dem nächsten Weg. Diesem kann anschließend bis zu einem Knoten gefolgt werden. Zu beachten ist lediglich, dass der gefundene Weg auch zu Fuß begehbar ist. Im obigen Beispiel erschließt sich dies anhand der Tatsache, dass der Startpunkt eine Haltestelle darstellt und davon ausgegangen werden kann, dass ein Bahnsteig existiert. Räumlich naheliegende Graphenknoten sind hingegen nur bedingt geeignet, da nicht bekannt ist, ob sie auch ohne weiteres vom Startpunkt aus erreichbar sind. So könnte sich, wie im obigen Beispiel, ein Schienenstrang oder auch eine Mauer zwischen Startpunkt und Startknoten befinden.

Berechnung der Route

Zur Lösung der Navigationsaufgabe eignen sich sowohl der Dijkstra- als auch der A* Algorithmus. Beiden ist gemein, dass sie den kürzesten Pfad in einem kantengewichteten Graphen finden.

Kostenfunktion Zunächst wird auf die Berechnung der Kantengewichte eingegangen. Die Gewichtung besteht zum einen aus der Länge des Wegstückes in Kilometer und zum anderen aus einem Faktor, der dessen Barrierefreiheit einschätzt. Letzteres ist nötig, da der kürzeste Weg vom Start zum Ziel nicht notwendigerweise optimal sein muss. Die Barrierefreiheit leitet sich von den Anforderungen Blinder und Sehbehinderter ab und ist nur teilweise auf andere Zielgruppen anwendbar. Die Skala für die Bewertung der Barrierefreiheit wurde in sechs Stufen aufgeteilt:

1. Sehr gut: Dazu zählen Tertiärstraßen und noch kleinere Nebenstraßen bzw. Straßen durch Wohngebiete. Sie zeichnen sich durch ein geringes Verkehrsaufkommen aus. Des Weiteren finden sich oft Bürgersteige zu beiden Straßenseiten. Straßennamen erleichtern die Orientierung und auftretende Kreuzungen sind übersichtlich aufgebaut und einfach zu überqueren.
2. Gut: Diese Kategorie beinhaltet alle Arten von befestigten Wegen, auf denen kein Verkehr herrscht. Darunter fallen beispielsweise asphaltierte oder betonierte Fuß- und Wanderwege. Solche Wege führen oft durch ruhige Gebiete und sind auf Grund der Bodenbeschaffenheit leicht zu begehen.
3. Neutral: Sekundärstraßen gelten auf Grund ihrer erhöhten Priorität im Straßennetz und dem damit verbundenen höheren Verkehrsaufkommen als neutral. Aber auch Fußgängerzonen werden dieser Kategorie zugeordnet, da die Navigation für Blinde auf Grund von im Weg stehenden Hindernissen oftmals erschwert wird. Des Weiteren ist der Straßenverlauf wegen fehlender Leitlinien wie Bordsteinkanten nicht uneingeschränkt nachverfolgbar.
4. Schlecht: Hierzu zählen vor allem Treppen und Privatwege. Treppen sind für Blinde und Sehbehinderte prinzipiell kein größeres Hindernis, benötigen allerdings die gesteigerte Aufmerksamkeit. Daher wird bei annähernd gleicher Länge statt

4. Entwurf

dessen eine Rampe bevorzugt. Anliegerstraßen werden oft als erstes oder letztes Wegsegment benötigt, da sie üblicherweise als Zufahrten zu Gebäuden dienen. Dennoch sind sie wegen ihrer beschränkten öffentlichen Nutzbarkeit eher zu meiden.

5. Sehr schlecht: Darunter fallen vor allem Primärstraßen und unbefestigte Fuß-, Wald- und Feldwege. Auf Primärstraßen herrscht der dichteste Verkehr und sie sind auf Grund komplexer Kreuzungssituationen schwer zu überqueren. Außerdem ist der Nutzer einem erhöhten Lärmpegel ausgesetzt.
6. Unpassierbar: Die letzte Kategorie sammelt alle, für Fußgänger nicht passierbaren Wegtypen. Dies sind vor allem Straßenbahn- und Eisenbahnschienen.

Als nächstes wird der Wichtungsfaktor bestimmt. Dieser entscheidet, wie hoch der Einfluss der Wegekategorie auf die Gesamtkosten ist. Dabei gilt:

$$\text{Länge eines Kategorie 1 Weges} = \text{Länge eines Kategorie 5 Weges} * \text{Faktor} \quad (4.1)$$

Der Faktor bestimmt also die Länge des Umweges, welche der Nutzer bereit ist, in Kauf zu nehmen. Während der Eine aber eher einen barrierefreien Weg bevorzugt, möchte ein Anderer lieber auf dem kürzesten Weg zum Ziel gelangen. Daher wird dem Nutzer die Gelegenheit gegeben, den Faktor selbst zu bestimmen.

Es folgt zunächst die Wichtungstabelle (Tabelle 4.2). Die Spalten stellen die Faktoren von 1 bis 4 dar. Die Zeilen stehen für die definierten Kategorien von 1 = sehr gut bis 5 = sehr schlecht.

	4x	3x	2x	1.5x	1x
1	60	50	33	20	0
2	30	25	16	10	0
3	0	0	0	0	0
4	-30	-25	-16	-10	0
5	-60	-50	-33	-20	0
10	200	200	200	200	200

Tabelle 4.2.: Tabelle zur Bestimmung der Routing Gewichte. Spalten: Umweg-Faktor, Zeilen: Wegklasse

Die Kostenfunktion für die Berechnung der Kantengewichte lautet:

$$c = l * (100 - w) \quad (4.2)$$

wobei:

- l: Länge des Wegstückes in km
- w: Gewicht aus obiger Tabelle bei gewähltem Faktor und Wegklasse

4. Entwurf

Ziel ist es, die Kosten c_{gesamt} über alle Wegknoten zu minimieren.

Dies soll anhand eines Beispiels verdeutlicht werden. Angenommen es existieren 3 Wege von Knoten A zu B:

- E_1 : Länge = 50 Meter, Kategorie 5 (primärstraße)
- E_2 : Länge 95 Meter, Kategorie 3 (Fußgängerzone)
- E_3 : Länge 130 Meter, Kategorie 1 (Straße durch ein Wohngebiet)

Wählt der Nutzer einen Umwegfaktor von 3, so ergeben sich die folgenden Streckenkosten:

- $cE_1 = 0.05 * (100 - (-50)) = 7.5$
- $cE_2 = 0.095 * (100 - 0) = 9.5$
- $cE_3 = 0.13 * (100 - 50) = 6.5$

Der günstigste und damit optimale Weg ist E_3 , obwohl er beinahe drei mal so lang ist wie E_1 . Wird hingegen der Faktor 1.5 gewählt, so ergeben sich die folgenden Werte:

- $cE_1 = 0.05 * (100 - (-20)) = 6.0$
- $cE_2 = 0.095 * (100 - 0) = 9.5$
- $cE_3 = 0.13 * (100 - 20) = 10.4$

Diesmal gewinnt eindeutig der Weg E_1 . Der Nutzer hätte lediglich einen Klasse 1 Weg akzeptiert, welcher 75 Meter lang und damit um die Hälfte länger als E_1 selbst gewesen wäre.

Der Faktor 2 und damit die Verdopplung der Weglänge bei Verbesserung der Wegklasse von sehr schlecht (5) zu sehr gut (1) bietet sich als Kompromiss an. Faktor eins ist äquivalent zur Suche des kürzesten Weges, da die Weglänge unabhängig von der Wegkategorie immer mit 100 multipliziert wird.

Die Wegkategorie 10 symbolisiert einen unpassierbaren Weg. Die Kosten sind unabhängig von der Weglänge immer kleiner als 0 und werden somit vom Routingalgorithmus ignoriert.

Zusätzlich zu dem Umwegfaktor erhält der Nutzer die Möglichkeit, einzelne Wege als unpassierbar einzustufen oder zu priorisieren. Ersteres ist beispielsweise nützlich, wenn sich auf der berechneten Route eine temporäre Baustelle befindet oder eine komplexe Kreuzung umgangen werden soll. Letzteres kann dazu dienen, eine dem Nutzer bekannte Straße entgegen der allgemeinen Einstufung zu bevorzugen. Die Einstufung muss sowohl im Vorhinein wie auch unterwegs möglich sein. Wird der Weg priorisiert, erfolgt die Einstufung als Wegklasse 1. Im Falle der Abwertung ist es ein Klasse 10 Weg. Dieser wird vom Routingalgorithmus wegen seines negativen Kantengewichts fortan ignoriert.

Des Weiteren ist der Ausschluss ganzer Wegklassen vom Routing vorgesehen. Es ist beispielsweise vorstellbar, dass Nutzer nicht über unbefestigte Wald- und Feldwege laufen und diese Wegklasse daher komplett vom Routing ausnehmen möchten. Gleiches gilt auch für Treppen.

Wahl des Routingalgorithmus Nachdem die Kostenfunktion für die Berechnung der einzelnen Kantengewichte definiert wurde, folgt nun die Wahl eines geeigneten Routingalgorithmus. In diesem Abschnitt werden der Dijkstra- und der A*-Algorithmus auf ihre Eignung hin untersucht.

Der Dijkstra-Algorithmus betrachtet von einem Knoten (A) aus alle möglichen Nachfolger und folgt demjenigen (B), welcher die geringsten Pfadkosten aufweist. Die aktualisierten Kosten für das Erreichen von B setzen sich aus den Kosten vom Startpunkt zum Knoten A plus den Kosten der Kante AB zusammen[9, Kapitel 3.4.2].

A* stellt eine Verallgemeinerung des Dijkstra-Algorithmus dar. Die Kostenfunktion wird um eine Heuristik erweitert, die die Entfernung des angepeilten Punktes (B) zum Zielpunkt abschätzt. Somit wird aus den potentiellen Nachfolgeknoten von A zuerst derjenige mit dem geringsten Abstand zum Ziel verfolgt[9, Kapitel 4.1.2].

Die Heuristik darf die Entfernung allerdings nicht überschätzen. Im konkreten Fall kommt die folgende Formel zum Einsatz:

$$h(x) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} * (100 - 60) \quad (4.3)$$

Erklärung:

- A und B stellen Punkte im zweidimensionalen Raum dar.
- Der Teil einschließlich der Wurzel berechnet die kürzeste Entfernung zwischen A und B. Die Krümmung der Erde kann auf Grund der geringen Entfernungen, wie sie bei der Fußgängernavigation auftreten, vernachlässigt werden.
- (100-60) ist der Wichtungsfaktor, mit dem die Distanz multipliziert werden muss, um die Kosten für das Wegstück zu erhalten. Da zum Definitionszeitpunkt der Heuristik nicht bekannt ist, welcher Wichtungsfaktor gewählt und welche Wegklasse betrachtet wird, ist bei beidem das Optimum anzunehmen. Anderenfalls könnte die Heuristik die Kosten überschätzen.

Sowohl der Dijkstra als auch der A* Algorithmus finden die optimale Lösung, also den Pfad mit den geringsten Kosten. Beide Algorithmen sind daher geeignet und stehen für das Routing zur Verfügung. Der Administrator des Routingsservers trifft die Wahl anhand der zu verwendenden Heuristik. Wird die weiter oben beschriebene verwendet, kommt der A* Algorithmus zum Einsatz. Ist hingegen $h(x) = 0$, so wird der Dijkstra Algorithmus zur Berechnung der Route genutzt. Die Auswahl der Heuristik geschieht durch einen Konfigurationsparameter, der beim Start des Servers übergeben wird.

Nachbearbeitung

Das Ergebnis des Routingalgorithmus aus dem vorherigen Abschnitt besteht zunächst lediglich aus einer Liste aufeinander folgender Wegsegmente inklusive Start- und Endknoten. Diese Liste muss im folgenden Schritt zu einer validen Route, wie in Kapitel 4.1 beschrieben, umgewandelt werden. Dabei ist zu beachten, dass redundante Teile ausgefiltert werden.

4. Entwurf

Begonnen wird mit einer leeren Route. Die Liste wird der Reihe nach abgearbeitet. Der nächste Knoten wird der Route hinzugefügt, wenn eine der drei Bedingungen zutrifft:

- Wenn die Route leer ist oder nur einen Knoten beinhaltet.
- Wenn sich zwischen dem letzten Segment der Route und dem Aktuellen eine Richtungsänderung ergibt. Dies kann entweder eine Kurve oder das Abbiegen an einer Kreuzung bedeuten.
- Wenn der letzte Routenpunkt eine wichtige Kreuzung ist.

Die Motivation hinter dem dritten Punkt bedarf weiterer Erläuterung. Dies soll anhand eines Beispiels geschehen. Angenommen, die erstellte Route führt den Nutzer einen langen Weg auf ein und derselben Tertiärstraße entlang, d.h. auf einem Kilometer Länge ist kein Abbiegevorgang nötig. Auf dem Weg zweigen allerdings immer wieder kleinere und größere Straßen ab. Die Aufgabe besteht nun darin, eine angemessene Anzahl von Kreuzungen in die Route aufzunehmen.

Eine Möglichkeit wäre, überhaupt keine geradeaus zu überquerenden Kreuzungen in die Routenbeschreibung zu integrieren. Dann bestünde die Route nur aus den Anweisungen:

- Folgen Sie der Straße A 1000 Meter geradeaus
- Biegen Sie anschließend in die Straße B ab

Diese knappe Beschreibung unterschlägt allerdings Kreuzungen mit anderen Haupt- und Nebenstraßen, auf denen Verkehr herrscht. So wird der Nutzer möglicherweise zu spät vor der Überquerung einer Straße gewarnt.

Ein weiterer Ansatz wäre, jede Abzweigung in der Routenbeschreibung zu erwähnen. Dies schließt allerdings auch kleine Wege, wie Trampelpfade und Fußwege mit ein. Auf einem 1000 Meter langen Wegstück können so schnell 20-30 Kreuzungen, die alle geradeaus überquert werden müssen und von denen meist keine Gefahr ausgeht, angesammelt werden. Selbstredend macht diese Überinformation die Route unübersichtlich und damit unbrauchbar.

Der Kompromiss sieht vor, dass eine Kreuzung als wichtig eingestuft wird, wenn auf mindestens zwei der einmündenden Straßen regelmäßig Verkehr herrscht und diese öffentlich nutzbar sind. Alternativ ist sie von Relevanz, wenn die Route Straßenbahn- oder Bahnschienen kreuzt.

Die Grafik 4.3 dient zu dessen Veranschaulichung. Sie zeigt die Nöthnitzer Straße im Süden von Dresden. Beispielhaft verläuft eine errechnete Route die Nöthnitzer Straße entlang von Osten nach Westen. Nach obiger Definition ist die Kreuzung Nöthnitzer Straße, Regensburger Straße in der Routenbeschreibung enthalten, da auf beiden Nebenstraßen Verkehr herrscht. Die darauf folgende, von der Nöthnitzer Straße nach Süden abzweigende Anliegerstraße wird hingegen ignoriert, da auf einer solchen Zufahrtsstraße, wenn überhaupt, dann nur selten Fahrzeuge unterwegs sind und diese beim Einbiegen auf die Nöthnitzer Straße auch auf querende Fußgänger Acht geben müssen.

4. Entwurf

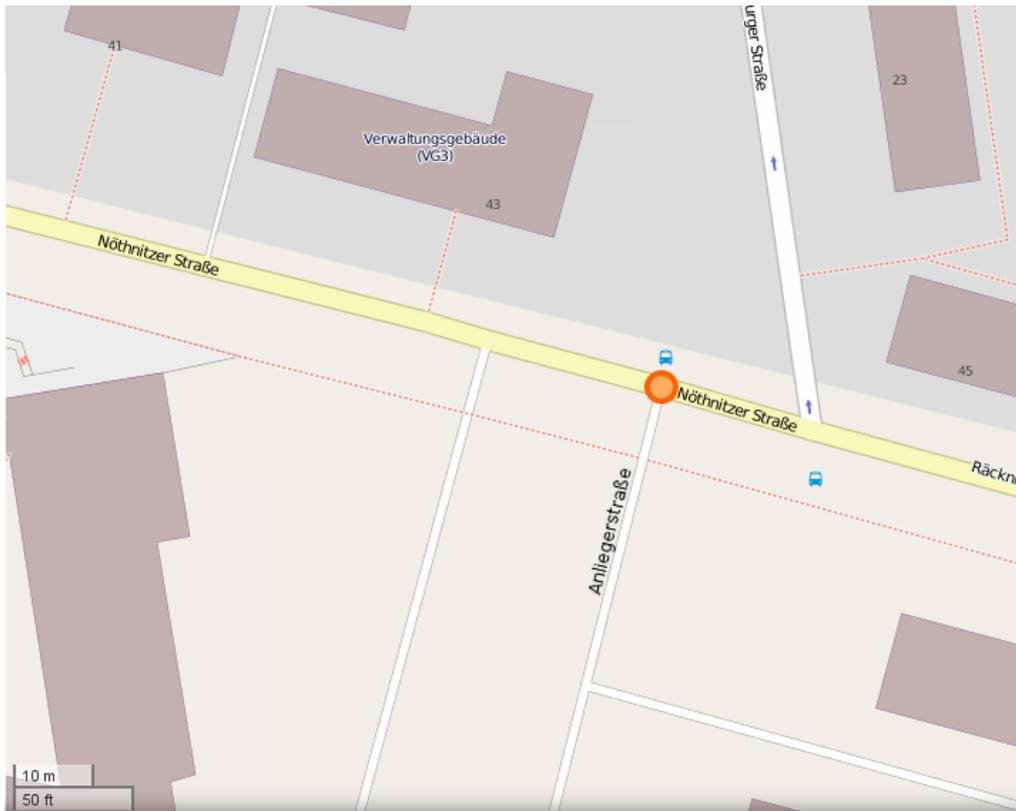


Abbildung 4.3.: Ein Kartenausschnitt zeigt die von Osten nach Westen verlaufende Nöthnitzer Straße im Süden von Dresden. Aus östlicher Richtung kommend zweigt zunächst die Regensburger Straße nach Norden ab. Etwas weiter in westlicher Richtung folgt die Abzweigung einer Anliegerstraße nach Süden. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

4.5.3. ÖPNV-Route

In diesem Abschnitt folgt eine konzeptionelle Beschreibung der Teilschritte, welche zur Integration des ÖPNV erforderlich sind (Anforderung A1.4 aus Kapitel 3.1). Dabei wird zuerst der Abruf der nächsten ÖPNV Routen erläutert. Anschließend folgt die Festlegung der Bewertungskriterien für die Auswahl der optimalen Verbindung.

Den Ausgangspunkt bilden natürlich weiterhin die Start- und Zielkoordinaten, welche im Abschnitt 4.5.1 gesammelt und an den Server übertragen wurden. Die Fahrplaninformationen stammen von der deutschen Bahn. Die DB stellt die Fahrpläne für alle Städte innerhalb Deutschlands zur Verfügung. Entgegen einiger lokaler Anbieter sind diese aber nicht in Echtzeit. Ein Nachteil, der in Kauf genommen wird, um nicht für jeden lokalen Betreiber eine eigene Datenschnittstelle implementieren zu müssen.

Das Interface der Deutschen Bahn stellt die folgenden Funktionen bereit:

4. Entwurf

- `Location = getLocation(name, type)`: Findet anhand eines namens und Typ das zugehörige Locations-Objekt. Der Typ kann eine Adresse oder eine Haltestelle sein.
- `Connections = calculateConnections(from, to, delay)`: Diese Funktion nimmt zwei Location Objekte entgegen und gibt eine Liste von Connections zurück. Mit dem delay Parameter kann die Abfahrtszeit nach hinten verschoben werden.
- `Departures = getDepartures(station)`: Listet die nächsten Abfahrten für eine gewählte Station auf.

Eine Connection, wie sie von der `calculateConnections` Funktion zurückgegeben wird, ist in Parts unterteilt. Ein Part kann eine zu Fuß oder per Bus/Bahn zurückzulegende Strecke darstellen. Der Part besteht aus Abfahrt, Strecke und Ankunft.

Dazu ein Beispiel. Connection: Dresden, Nöthnitzer Straße 46 -- Dresden, Görlitzer Straße 2, Part 1 von 4:

1. Abfahrt: Dresden, Nöthnitzer Straße 46 (Adresse), 8 Minuten Fußweg, Ankunft: Dresden, Münchner Platz (Haltestelle)
2. Abfahrt: Dresden, Münchner Platz (Haltestelle), Fahrt mit Linie T3 (Wilder Mann), Ankunft: Dresden, Albertplatz (Haltestelle)
3. Abfahrt: Dresden, Albertplatz (Haltestelle), Fahrt mit Linie T7 (Weixdorf), Ankunft: Dresden, Louisenstraße (Haltestelle)
4. Abfahrt: Dresden, Louisenstraße (Haltestelle), 12 Minuten Fußweg, Ankunft: Dresden, Görlitzer Straße 2 (Adresse)

Zuerst werden zu den Start- und Zielkoordinaten die jeweiligen Adressen ermittelt. Anschließend stellt der Server eine Verbindungsanfrage, bestehend aus Start- und Zieladresse an die API der Deutschen Bahn. Die Antwort beinhaltet die nächsten vier Verbindungen. Im nächsten Schritt muss aus der erhaltenen Liste die optimale Verbindung ausgewählt werden. Dazu wird im Unterabschnitt 4.5.3 eine Bewertungsfunktion entworfen, welche für eine gegebene Verbindung deren Kosten errechnet.

In einigen Fällen ist keine der vier Verbindungen optimal. Gründe hierfür sind u.a. zu knapp bemessene Umstiegszeiten oder eine hohe Anzahl an Umsteigevorgängen. Dann müssen alternative Verbindungen abgerufen und evaluiert werden. Dies gilt auch, falls den Start- oder Zielkoordinaten keine Adresse zugewiesen werden kann, beispielsweise wenn sich der Nutzer abseits der Straßen auf Wald- oder Feldwegen bewegt. Alternativ wird auch nach zusätzlichen Verbindungen gesucht, falls unter den Bisherigen noch keine Direktverbindung vorhanden war.

Dazu werden ausgehend vom Start- und Zielpunkt alle umliegenden Haltestellen im Umkreis von 500 Metern ermittelt. Diese Entfernung entspricht etwa einem 10 bis 15 minütigen Fußweg und stellt für die meisten Nutzer eine zu bewältigende Strecke dar. Bei einer größeren Anzahl von in Frage kommenden Haltestellen werden aus Performancegründen nur die vier Naheliegendsten verwendet. Wurde jedoch keine Haltestelle gefunden, muss der Suchradius sukzessive vergrößert werden.

4. Entwurf

Anschließend wird von jeder der Starthaltestellen eine Verbindung zu jeder Zielhaltestelle abgefragt. Jede Abfrage enthält erneut die vier nächsten Einzelverbindungen. Somit müssen zusätzlich maximal $4 * 4 * 4 = 64$ Einzelverbindungen bewertet werden. Die Verbindung, welche die geringsten Kosten aufweist, wird als Grundlage für die Berechnung der Route verwendet. Alternativ ist auch denkbar, dem Nutzer die besten drei Verbindungen vorzuschlagen und ihn wählen zu lassen. Dies ist besonders für Ortskundige von Vorteil.

Der nächste Schritt besteht in der Routenberechnung für die zu Fuß zurückzulegenden Teilabschnitte der Verbindung. Angewendet auf das Beispiel von Seite 30 bedeutet das, dass Part 1 und 4 mit der aus Kapitel 4.5.2 erläuterten Methode bestimmt und dem Nutzer zusammen mit Part 2 und 3 als fertige Route präsentiert werden. Abschließend erfolgt die Übermittlung der zusammengefügten Route an das Endgerät.

Bewertungskriterien

Dieser Abschnitt listet die Anforderungen an eine optimale Verbindung des ÖPNV auf und legt eine Bewertungsfunktion fest.

Über die Anforderungen, welche Blinde und Sehbehinderte an eine optimale ÖPNV Verbindung stellen, konnte im Zuge der Recherche wenig in Erfahrung gebracht werden. Daher werden die nachfolgenden Hauptanforderungen anhand der bisher zusammengetragenen Informationen abgeleitet.

Bereits in Kapitel 1 wurde festgestellt, dass die Gruppe der Blinden substantielle Einschränkungen bei der Orientierung in unbekanntem Gebieten erfährt. Dies liegt in der eingeschränkten Wahrnehmungsfähigkeit begründet und führt zu einer erhöhten mentalen Belastung während des Navigationsvorgangs. Dabei gilt: je komplexer die zu erkundende Umgebung, desto anstrengender ist der Orientierungsprozess. Als komplex werden beispielsweise weitläufige Plätze und Kreuzungen sowie große Bahnhöfe und Mehrfachhaltestellen eingestuft [17, Kapitel 3.4.1 und 3.5.1].

Ziel ist daher, diejenige Verbindung zu ermitteln, bei der die mentale Belastung minimiert wird. Dazu muss zunächst geklärt werden, welche Teile der Route am anstrengendsten und somit am schwersten zu bewältigen sind.

Die Abschnitte, welche fahrend zurückgelegt werden, bilden hierbei die am wenigsten anstrengenden Teilstücke der Route, da der Nutzer während der Fahrt passiv bleiben kann. Die Teilstücke vom Startpunkt zur Starthaltestelle und von der Zielhaltestelle bis zum Zielpunkt verlangen dem Nutzer hingegen ein erheblich höheres Maß an Konzentration ab. Das begründet sich daraus, dass die Strecken zu den Haltestellen oftmals durch dem Nutzer unbekannte Gebiete führen und er sich dessen Verlauf zunächst mittels geeigneter Hilfsmittel erarbeiten muss.

Auch das Umsteigen trägt signifikant zur mentalen Belastung bei. Zum Einen muss das korrekte Anschlussverkehrsmittel gefunden werden, d.h. die Liniennummer bzw. das Fahrziel ist in Erfahrung zu bringen. Dies wird erschwert, wenn an der betreffenden Haltestelle mehrere Fahrzeuge hintereinander halten können, da auf diese Weise eines davon eventuell nicht wahrgenommen wird. Erfolgen während der Fahrt keine Durchsagen des nächsten Haltestellennamens, so muss ferner das Erreichen der richtigen Zielhaltestelle

erfragt werden.

Nachdem der Nutzer an der Umsteigehaltestelle aus dem ankommenden Fahrzeug ausgestiegen ist, muss er die Position des abfahrenden Verkehrsmittels aufsuchen. Dieser Umsteigevorgang ist bezüglich seiner Komplexität wie folgt zu differenzieren: Die einfachste Form stellt die Weiterfahrt auf dem selben Bahnsteig bzw. Haltestelle dar. In diesem Fall kann der Nutzer nach dem Aussteigen einfach stehen bleiben und auf den Anschluss warten.

Bei der nächst komplexeren Variante muss er zur gegenüberliegenden Haltestelle bzw. Bahnsteig wechseln. Im Fall von Bus und Straßenbahnen bedeutet dies das Überqueren der Straße bzw. Straßenbahngleise. Ein weitreichenderer Ortswechsel ist allerdings nicht nötig, da sich die Haltestellen in der Regel entweder direkt gegenüber oder um wenige Meter versetzt befinden. Um diese zwei Varianten allerdings unterscheiden zu können, müssen für jede Haltestellenposition alle abfahrenden Linien inklusive Fahrziel bekannt sein. Der Abschnitt 4.5.3 geht genauer auf die Ermittlung der exakten Haltestellenposition ein. Fehlen diese Daten kann lediglich die Aussage getroffen werden, ob die Weiterfahrt von der gleichen Haltestelle fortgesetzt werden kann.

Die dritte Möglichkeit ist, dass der Nutzer den ankommenden Haltestellenbereich verlassen muss, um den Umsteigevorgang durchzuführen. Dies ist beispielsweise beim Umsteigen vom Bus zur S-Bahn aber auch bei großen Bus- oder Straßenbahnhaltestellen der Fall. Ein konkretes Beispiel für letztere stellt der Dresdner Albertplatz dar. Dessen Haltebereich besteht aus je zwei Haltestellenpaaren, welche etwa 100 Meter voneinander entfernt sind. Diese Umstiegsklasse fordert die höchste Konzentration denn oftmals liegen die betreffenden Haltestellenbereiche an Verkehrsknotenpunkten und zum Erreichen der Abfahrtsstation muss mehr als eine Straße bzw. Gleis überquert werden. Daher ist für einen solchen Umsteigevorgang ein gewisses Maß an Zeit einzuplanen.

Innerhalb dieser zuletzt definierten Kategorie gilt es, nochmals zwischen zwei Fällen zu unterscheiden. Bei einigen Verkehrsmitteln liegt der Abfahrtsort abseits von Straßen und Fußwegen und zusätzlich oftmals innerhalb eines Gebäudes. Zu diesen Verkehrsmitteln gehören sowohl S- und U-Bahnen wie auch Regionalzüge. Da derzeit noch kein flächendeckendes Indoor-Routing zur Verfügung steht, ist bei der Beteiligung von einem dieser Verkehrsmittel keine exakte Wegbeschreibung bis zum Gleis, sondern nur bis zum Eingang des Gebäudes möglich. Für die Orientierung innerhalb des Gebäudes ist entweder eine grundlegende Kenntnis von dessen Aufbau nötig, oder es muss die Hilfe von Passanten erbeten werden.

Aus diesen Gründen wurden die unterschiedlichen Verkehrsmitteltypen in zwei Klassen aufgeteilt:

- Klasse A umfasst Stadt- und Fernbusse sowie Straßenbahnen
- Klasse B umfasst S- und U-Bahnen sowie Regionalbahnen

Weitere Transportmittel wie beispielsweise Fähren werden derzeit nicht berücksichtigt, können aber jederzeit zu einer der Klassen hinzugefügt werden. Bildet ein Verkehrsmittel der Klasse B einen Teilabschnitt der Verbindung, so ist der Umsteigevorgang auf Grund

4. Entwurf

der teils fehlenden Routinganweisungen innerhalb der Gebäude gegenüber Klasse A nochmals geringfügig abzuwerten.

Zusammengefasst bilden die zu Fuß zurückzulegenden Wege und die Umsteigevorgänge die mental anstrengenden Routenteile, während die Fahrt mit einem Verkehrsmittel nur geringfügig ins Gewicht fällt. Daher ist bei der Bewertung einer ÖPNV Verbindung zwischen der Länge des Fußweges und der Umstiege abzuwägen. Dabei ist die Komplexität des jeweiligen Umsteigevorgangs zu berücksichtigen.

Ein Umstieg, bei welchem der Nutzer von der selben Haltestelle weiterfährt, ist einem mehrere hundert Meter langen Fußweg zumeist vorzuziehen. Muss hingegen die Verkehrsmittelklasse beim Umsteigen gewechselt werden, so erscheint der längere Fußweg als die bessere Option.

Die folgende Aufzählung definiert die Kriterien für eine Bewertungsfunktion zur Abwägung der verschiedenen Faktoren einer ÖPNV Verbindung:

- 1 Punkt für 100 Meter Fußweg. Die zurückzulegende Strecke vom Startpunkt zur Haltestelle, beim Umsteigen oder von der Haltestelle zum Zielpunkt wird lediglich geschätzt, indem die kürzeste Entfernung zwischen den Punkten (Luftlinie) Verwendung findet.
- 1 Punkt für 5 Minuten Wartezeit: Dies gilt nur für die Zeit bis zum Verbindungsstart, d.h. für die Differenz zwischen aktueller Uhrzeit und der Abfahrt an der Starthaltestelle.
- 1 Punkt für 10 Minuten Fahrzeit: Zeitdifferenz zwischen Ziel- und Starthaltestelle inkl. Wartezeit bei Zwischenaufenthalten.
- Umstiege, wenn die genaue Position von Start- und Zielhaltestelle bekannt sind:
 - 2 Punkte: Gleicher Bahnsteig d.h. der Fahrgast muss nach dem Aussteigen lediglich stehen bleiben und auf die nächste Bahn warten.
 - 4 Punkte: Gegenüberliegender Bahnsteig bzw. bei Bus oder Straßenbahn die gegenüberliegende Haltestelle. Dies gilt allerdings nur, wenn mindestens 5 Minuten zum Umsteigen bleiben.
 - 6 Punkte: Anderer Bahnsteig oder Haltestelle, ausschließlich Verkehrsmittelklasse A bei mindestens 10 Minuten Umsteigezeit.
 - 8 Punkte: Anderer Bahnsteig oder Haltestelle, Beteiligung von Verkehrsmittelklasse B bei mindestens 10 Minuten zum Umsteigen.
- Umstiege, falls die genaue Position mindestens einer Haltestelle nicht bekannt ist:
 - 4 Punkte: Bei der gleichen Station und mindestens 5 Minuten Zeit zum Umsteigen.
 - 6 Punkte: Bei verschiedenen Stationen, ausschließlich Verkehrsmittelklasse A und mindestens 10 Minuten fürs Umsteigen.

4. Entwurf

- 8 Punkte: Bei verschiedenen Stationen, Beteiligung der Verkehrsmittelklasse B und mindestens 10 Minuten Umsteigezeit.

Bei Nichteinhaltung der Mindestumstiegszeit addiert der Algorithmus zu der bisherigen Punktzahl zusätzliche 100 Punkte. Die betreffende Verbindung würde nur dann gewählt, wenn überhaupt keine Alternative mit adäquaten Umsteigezeiten existiert.

Genauere Haltestellenposition

In diesem Abschnitt werden Maßnahmen zum Auffinden der genauen Haltestellenposition diskutiert. Zunächst wird die Problemstellung erläutert, bevor im zweiten Teil auf die Lösungsansätze eingegangen wird.

Eine ÖPNV Haltestelle umfasst zumeist ein größeres Areal. Bei Bus- und Straßenbahnhaltestellen existieren beispielsweise fast immer mindestens zwei Haltepositionen, um beide Fahrtrichtungen abzudecken. Gelegentlich können es aber auch wesentlich mehr sein. Die Bushaltestelle Dresden Helmholzstraße umfasst z.B. genau zwei Haltepositionen, zum Einen in Richtung Striesen und zum Anderen in Richtung Löbtau. Die Haltestelle Dresden Cämmerswalder Straße besteht hingegen aus 4 Positionen, zwei für die Straßenbahnlinie 3 (stadtein- und stadtauswärts) und zwei für den Bus (Linie 63) nach Löbtau und Pillnitz (siehe Abbildung 4.4). Bei S- und U-Bahnstationen sind hingegen oftmals mehrere Gleise bzw. Ebenen vorhanden.

Die eingangs betrachteten Navigationssysteme unterscheiden, insofern sie ein Routing anbieten, allerdings nicht zwischen den Stoppositionen, sondern belassen es bei den Haltestellennamen. Dies ist insbesondere bei großen Haltestellen auf Grund der hohen Ungenauigkeit problematisch und sollte weitestgehend vermieden werden.

Die OSM beinhaltet für die allermeisten Haltestellen die jeweiligen Haltepositionen. Wird Dresden als Beispiel herangezogen, so fand sich bisher jede überprüfte Haltestelle in der Datenbank. Zusätzlich sind die Routen der Bus- und Bahnlinien verzeichnet. Eine Route besteht aus Name und Richtung der Linie, den Streckenabschnitten und den anzufahrenden Haltestellen.

Betrachtet wird zunächst die Situation bei Bus und Straßenbahn. Gegeben ist eine Verbindung von der Start- zur Zielhaltestelle, wie sie im vorigen Abschnitt abgefragt wurde. Der Einfachheit halber findet erneut die obige Beispielverbindung von der Nöthnitzer Straße 46 zur Görlitzer Straße 2 Verwendung. Um die Route für den ersten zu Fuß zurückzulegenden Routenabschnitt berechnen zu können, wird dessen Zielposition benötigt.

Die Haltestelle Münchner Platz besteht aus zwei Stoppositionen. Zunächst wird für jede der Positionen eine Liste aller ankommenden Linien erstellt. Das Ergebnis lautet folgendermaßen:

- Münchner Platz, stadtauswärts: [Linie 3 nach Coschütz]
- Münchner Platz, stadteinwärts: [Linie 3 nach Wilder Mann]

In diesem Fall hält jeweils nur eine Linie an der Haltestelle. Nun wird versucht, eine Übereinstimmung mit der Fahrtanweisung aus Routenabschnitt 2 zu finden („Fahrt mit

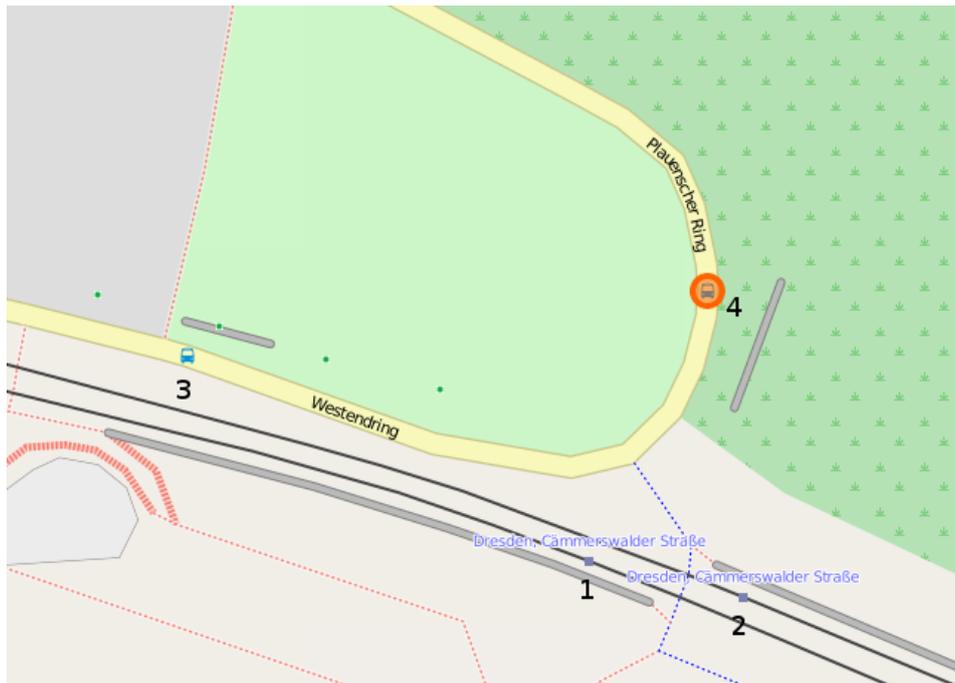


Abbildung 4.4.: Das Bild zeigt das Haltestellenareal der Dresdner Bus- und Straßenbahnhaltestelle Cäpperswalder Straße. Die Punkte 1 und 2 markieren die Straßenbahnhaltestellen. Die Ziffern 3 und 4 stellen die Bushaltestellen dar. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

Linie T3 nach Wilder Mann“). In diesem Beispiel existiert eine Übereinstimmung mit der stadteinwärts gelegenen Haltestelle. Somit kann die genaue Haltestellenposition und damit der Zielpunkt der Fußgängeroute exakt ermittelt werden. Analog verhält es sich beim Umsteigen auf dem Albertplatz.

Nach ersten Tests stellte sich aber heraus, dass zwar die Haltestellenpositionen alle in der OSM verzeichnet sind, es jedoch teilweise am Routenverlauf der Bus- und Bahnlinien mangelt. So fehlt es besonders bei den Buslinien an der Fahrtrichtung. Wäre für das Beispiel des Münchner Platzes lediglich bekannt, dass in beiden Richtungen die Linie 3 fährt, so könnte nicht ermittelt werden, zu welcher der Stoppositionen der Nutzer geroutet werden soll. In einem solchen Fall ist die genaue Haltestellenposition nicht ermittelbar und der Nutzer muss bei Erreichen des Haltestellenareals darauf hingewiesen werden.

Die gleichen Einschränkungen ergeben sich auch beim Umsteigen. Wenn bei der Ankunfts- oder AbfahrtsHaltestelle die genaue Haltestellenposition nicht bestimmt werden kann, ist keine Aussage über den Umsteigeverlauf mehr zu tätigen. In diesem Fall bleibt nur noch die Unterscheidung verschiedener Haltestellen, anhand der von der Deutschen Bahn mitgelieferten Stations-Id.

Fehlt die genaue Haltestellenposition, so wirkt sich das auch auf die Genauigkeit

der Fußgängeroute aus. Konnte die Zielhaltestelle nicht genau bestimmt werden, so stimmen die ersten Routinganweisungen eventuell nicht überein, da bei der Berechnung der Route eine falsche Haltestellenposition verwendet wurde. Der Nutzer kann diese Ungenauigkeiten aber noch recht einfach kompensieren, indem er direkt zur nächsten oder ggf. übernächsten Anweisung springt. Somit erreicht er ausgehend von der aktuellen Position mit Hilfe des Kompass sicher einen der nächsten Fixpunkte, beispielsweise eine Kreuzung. Sehr ungünstig ist das Fehlen der genauen Haltestellenposition wenn diese als Starthaltestelle dient und somit zu Fuß zu erreichen ist. In diesem Fall muss eventuell fremde Hilfe in Anspruch genommen werden.

Bei Haltestellen der Verkehrsmittelklasse B stehen generell weniger Informationen zur Verfügung. Im Wesentlichen ist dies die Gleisnummer und die Art des Verkehrsmittels. Nach derzeitiger Datenlage ist das Auffinden der genauen Position nicht möglich. Das liegt meistens daran, dass der eigentliche Bahnsteig überdacht oder unter der Erde liegt und ein Indoor Routing, wie bereits angesprochen, bisher flächendeckend nicht verfügbar ist. Wenn die genaue Position des Gebäudeeingangs bekannt ist, erfolgt das Routing bis zu diesem Punkt und informiert den Nutzer anschließend über die aufzusuchende Gleisnummer. Anderenfalls kann die Bahnhofposition nur ungefähr mitgeteilt werden.

4.5.4. Darstellung der Route auf dem Client

Dieser Abschnitt erläutert den letzten Teil des Routings: Die Verfolgung der erstellten Route auf dem Clientsystem. Einführend wird ein grober Überblick über den Aufbau von dessen Benutzungsoberfläche gegeben. In dem sich anschließenden Abschnitt werden Ansätze zur Verbesserung der Positionsbestimmung diskutiert. Schließlich folgt in einem weiteren Abschnitt ein Entwurf zur Erkennung der Straßenseite.

Aufbau der Benutzungsoberfläche

Gemäß den Anforderungen aus A3.3 (Kapitel 3.1) wurde eine statische Menüleiste mit den folgenden Einträgen im oberen Bereich des Hauptfensters platziert:

- Start: In diesem Bereich werden Start- und Zielpunkt der Route sowie einige weitere Routenoptionen ausgewählt (siehe 4.5.1). Diese Ansicht dient als Startseite des Programms.
- Router: Dieser Tab stellt die empfangene Route dar.
- POI: Hier werden Points of Interest aus der Umgebung angezeigt. Die Auswahl kann durch die Wahl verschiedener POI Kategorien eingeschränkt werden. So sind beispielsweise Kategorien für ÖPNV Haltestellen, Restaurants oder Kreuzungen denkbar. Das Anlegen der Kategorien obliegt dem Nutzer. Der Übersichtlichkeit halber ist auch die richtungsabhängige Filterung der POI Liste möglich. Ist diese ausgewählt, werden dem Nutzer lediglich vor ihm liegende POI angezeigt.

4. Entwurf

Anhand der Menüeinträge wechselt der Nutzer zwischen den Hauptbereichen der Benutzungsoberfläche. Daneben existieren einige weitere Oberflächenbereiche wie beispielsweise Dialoge zur Anzeige des GPS Status und der Eigenschaften eines Routenobjekts.

Der Router Tab und der Dialog zur Anzeige der Routenobjektdetails werden zur Darstellung der geladenen Route benötigt und in den folgenden Abschnitten ausführlicher beschrieben.

Router Tab Nachdem die im vorherigen Kapitel erstellte Route vom Client empfangen und verarbeitet wurde, wechselt die Ansicht automatisch und zeigt den Router Tab an. Dieser Tab gliedert sich nochmals in zwei Untermenüpunkte: Routenverfolgung und Routenliste.

Die Routenliste bietet einen Überblick über die gesamte Route. Jedes Routenelement wird aus Platzgründen lediglich anhand seiner wichtigsten Eigenschaften beschrieben. Bei einer Kreuzung sind dies beispielsweise deren Name, die Anzahl der Straßen und die Routinganweisung. Es ist aber jederzeit möglich, sich sämtliche Objektattribute auf einer Detailseite anzeigen zu lassen. Das mit einem „*“ markierte Listenobjekt symbolisiert die aktuelle Routenposition (siehe Abbildung 4.5).

In der Routenverfolgungsansicht findet die Navigation von Punkt zu Punkt statt. Der Bereich ist dreigeteilt:

Oben stehen statische Informationen, welche das nächste Wegstück und den darauffolgenden Wegpunkt beschreiben. Diese beinhalten u.a. Namen und Länge des nächsten Segments sowie die anschließende Routinganweisung.

Die Mitte bilden dynamische Informationen über Richtung und tatsächliche Entfernung des nächsten Routenpunktes. Diese Informationen werden aktualisiert, sobald eine neue GPS Position oder ein abweichender Kompasswert vorliegen.

Die Richtung ergibt sich aus der Differenz der Himmelsrichtung des nächsten Segments und dem aktuellen Kompasswert. Beide Werte liegen in Grad von 0 bis 359 vor. Ist das Ergebnis kleiner 0, so wird 360 dazu addiert, um einen positiven Wert zu erhalten. Anschließend folgt der besseren Verständlichkeit halber die Konvertierung in ein Richtungsformat, welches an das Ziffernblatt einer Analoguhr angelehnt ist. So entsprechen:

- Geradeaus: $0^\circ = 12$ Uhr
- Rechts: $90^\circ = 3$ Uhr
- Links: $270^\circ = 9$ Uhr

Sobald der Nutzer in die Nähe einer Kreuzung gelangt, werden außerdem Richtung und Name der zugehörigen Straßen angezeigt. Die Daten passen sich dabei der jeweiligen Blickrichtung des Nutzers an und dienen zur besseren Orientierung im Kreuzungsbereich.

Unten im Display befinden sich schließlich die Buttons zur Steuerung. Über den Details Button lassen sich auch an dieser Stelle weitere Informationen über den nächsten Wegpunkt anzeigen. Die benachbarten Schalter dienen zur Wahl des vorherigen bzw. nächsten Routenpunktes. Mit Hilfe des „Markers“ (ganz rechts) kann die aktuelle Position

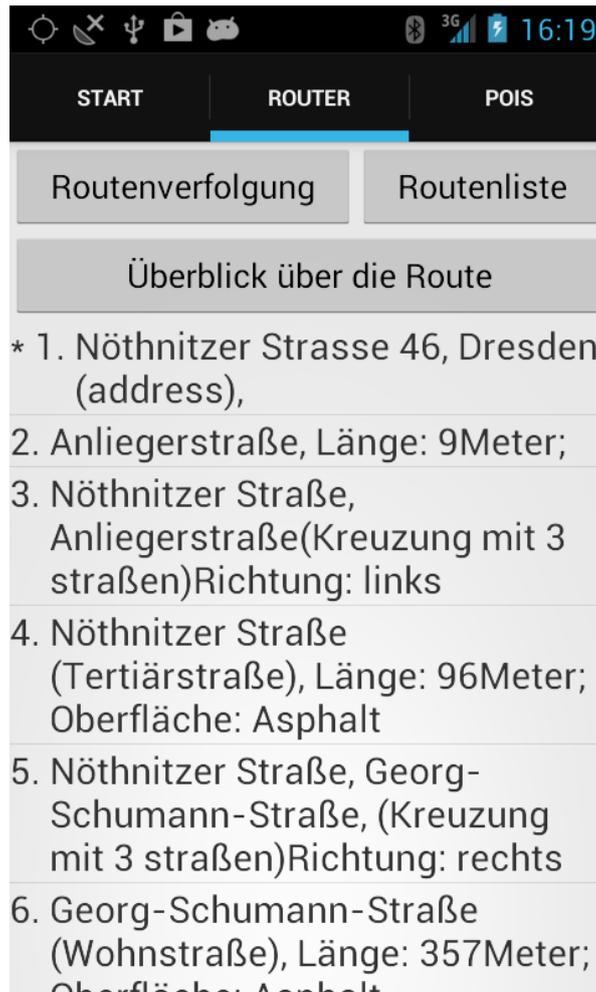


Abbildung 4.5.: Screenshot des prototypischen Android Clients. Eine Route wurde geladen und der Router Tab ist ausgewählt. Die ersten Routinganweisungen werden untereinander in einer Liste dargestellt.

bei Bedarf gespeichert und später wieder aufgefunden werden. Aus der daruntergelegenen Combobox kann der Nutzer eine POI Kategorie auswählen. Danach wird er während der Routenverfolgung über in der Nähe liegende POI dieser Kategorie informiert.

Abbildung 4.6 zeigt einen Screenshot der Routenverfolgungsansicht.

Details-Dialog Dieser Dialog dient zur Anzeige aller Attribute eines Routenobjekts. Bei einfachen Wegpunkten sind dies der Name, die Entfernung und Richtung sowie die geographischen Koordinaten.

Bei einer Kreuzung kommt außerdem eine Liste aller Straßen bzw. Wege inklusive deren Namen, Typ und Richtung hinzu. Die Richtungsangaben hängen von der Entfernung des Nutzers zur Kreuzung ab. Befindet er sich in unmittelbarer Kreuzungsnähe, erfolgen

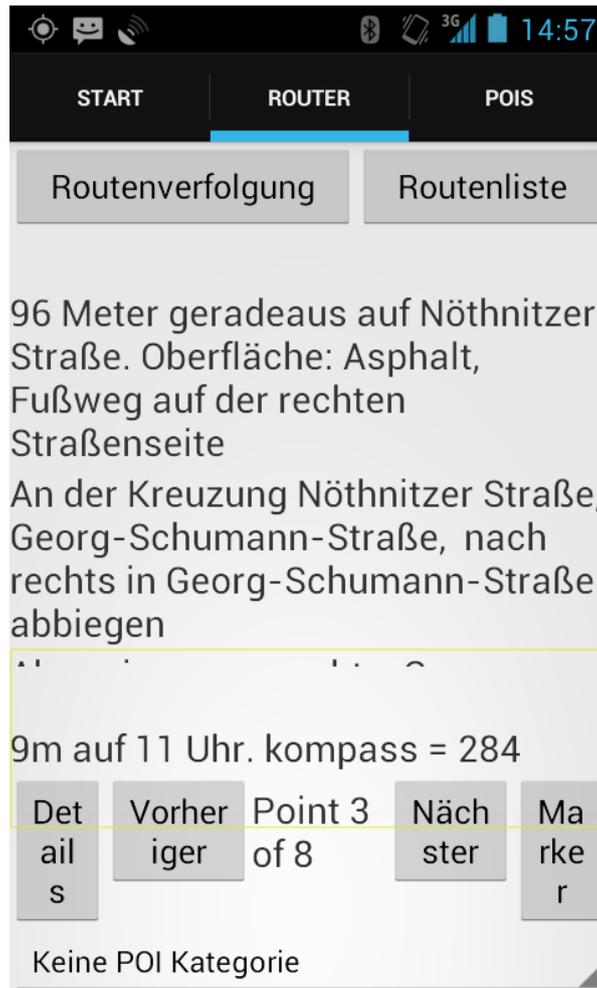


Abbildung 4.6.: Screenshot des prototypischen Android Clients. Eine Route wurde geladen und der Router Tab ist ausgewählt.

die Richtungsangaben abhängig von den Werten des internen Kompass. Sonst werden die statischen Himmelsrichtungen verwendet. Des Weiteren werden, sofern vorhanden, alle Ampelpfosten ebenfalls nach Entfernung und Richtung sortiert dargestellt.

Points of Interest bestehen hingegen aus der Adresse und den Kontaktdaten wie Telefonnummer, E-Mail Adresse und Webseite sowie Angaben zu Öffnungszeiten und der Barrierefreiheit der Lokation. Des Weiteren ist abzulesen, ob sich der Punkt innerhalb eines Gebäudes befindet oder selbst eines darstellt. Falls ja, werden neben dessen Namen auch die Position der Gebäudeeingänge angezeigt.

Eine Haltestelle des ÖPNV bzw. ein Bahnhof erweitert die Eigenschaften eines POI noch um die Liste aller abfahrenden Verkehrsmittel. Dazu kommt der aktuelle Haltestellenfahrplan, welcher die nächsten Abfahrten auflistet.

Verbesserung der Positionsgenauigkeit

Das GPS Signal unterliegt ein paar Einschränkungen, die sich negativ auf die Positionsgenauigkeit auswirken. Nachfolgend werden einige davon beschrieben und Lösungsansätze aufgezeigt.

Die erstmalige Bestimmung der Position nach dem Programmstart benötigt zu viel Zeit. Bis die Position ermittelt wurde, können teils mehrere Minuten vergehen. Innerhalb von Gebäuden ist häufig überhaupt keine Positionsbestimmung möglich. Letzteres erschwert die Auswahl von Start und Ziel, solange sich der Nutzer beispielsweise noch zu Hause befindet (Anforderung A1.3). Zur Verbesserung der Situation wird das Assisted Global Positioning System (A-GPS) Verfahren eingesetzt.

Dabei erfolgt die Ortung zusätzlich über einen zweiten vom GPS Signal unabhängigen Kanal. So verrät die Funkzelle, in welche der Client eingebucht ist, dessen ungefähre Position. Diese wird durch Messungen der Signallaufzeiten von benachbarten Mobilfunkmasten noch präzisiert. Ist das Smartphone mit einem Wifi Netzwerk verbunden, so kann dessen Position in einer Datenbank nachgeschlagen werden. Eine auf diese Weise ermittelte Position genügt in den meisten Fällen für das Anzeigen der Points of Interest in der Umgebung. Mit A-GPS wird auch die initiale GPS Position beim Verlassen des Gebäudes innerhalb von Sekunden gefunden und der Benutzer kann ohne größere Verzögerungen mit dem Ablaufen der berechneten Route beginnen[6].

Jede empfangene GPS Position enthält neben Längen- und Breitengrad noch weitere Attribute wie dessen Empfangsdatum und einen Genauigkeitswert in Metern. Durch äußere Einflüsse wie beispielsweise die Nähe zu hohen Gebäuden unterliegen die stetig empfangenen GPS Signale allerdings gewisser Qualitätsschwankungen. Bevor die nächste ermittelte Position an die Anwendung übergeben wird, muss daher eine Vorfilterung stattfinden. Dazu werden zuerst die Genauigkeitswerte der letzten akzeptierten und der neuen Position verglichen. Die neue Position wird übernommen, wenn deren Genauigkeitswert unterhalb eines definierten Schwellwertes liegt. Alternativ findet die neue Position Verwendung, falls sie trotzdem noch genauer ist als die zuletzt Akzeptierte. Trifft beides nicht zu, wird sie dennoch übernommen, falls die alte Position bereits über 10 Sekunden alt ist. So wird eine Überalterung der Standortdaten vermieden. Wenn sich das GPS Signal drastisch verschlechtert oder für eine gewisse Zeit ganz ausfällt, ist der Nutzer hierüber in Kenntnis zu setzen.

Trotz der Vorfilterung liegt die maximale GPS Genauigkeit bei etwa 10 Metern (siehe Kapitel 2). Für die Verfolgung eines Weges ist dieses Maß an Ungenauigkeit tollerabel. Soll jedoch ein konkreter Routenpunkt wie beispielsweise eine Kreuzung oder ein Gebäudeeingang aufgesucht werden, so ist das mit diesem Maß an Ungenauigkeit nur eingeschränkt möglich. Daher wird in den folgenden Absätzen ein Verfahren entwickelt, welches den Nutzer beim Auffinden von Routenpunkten unterstützt.

Dazu wird zum Einen die Richtung eines Routensegments benötigt. Dessen Richtungswert ist statisch und gibt den Verlauf einer Wegstrecke vom Start- zum Zielpunkt des Segments in Grad an. Zum Anderen wird fortlaufend die Richtung zwischen der aktuellen Nutzerposition und dem Segmentzielpunkt berechnet. Der zwischen diesen zwei Segmenten aufgespannte Winkel wird nun wie folgt eingesetzt, um das Erreichen des

4. Entwurf

Segmentzielpunktes zu erkennen.

Gegeben sind zwei Punkte A und B einer Route (siehe Abbildung 4.7 auf Seite 41). Das Wegsegment zwischen den Punkten wird als \overline{AB} bezeichnet und verläuft annähernd gradlinig. Der Nutzer läuft das Teilstück der Route von A nach B ab. Nun wird die Richtung R_{AB} des Segments errechnet. Dieser Wert kann im Voraus berechnet werden und ändert sich nicht mehr.

Der Nutzer startet bei Punkt A und bewegt sich entlang des Wegsegments zu Punkt B. Die Punkte X_1 , X_2 und X_3 stellen empfangene GPS Koordinaten zu verschiedenen Zeitpunkten dar. α_1 bis α_3 stehen für die Winkel, die von \overline{AB} und den jeweiligen Strecken $\overline{X_1B}$ bis $\overline{X_3B}$ eingeschlossen werden. Sie berechnen sich folgendermaßen:

$$\alpha = \text{Min}(|R_{AB} - R_{XB}|, 360 - |R_{AB} - R_{XB}|) \quad (4.4)$$

R_{AB} und R_{XB} bezeichnen die Himmelsrichtungen der jeweiligen Strecken in Grad. Die Minimum-Funktion garantiert, dass ausschließlich der Innenwinkel berechnet wird. Die Werte für die Strecke \overline{XB} und den Winkel α werden fortlaufend aktualisiert.

Der Verlauf des Winkels α soll nun beim Abschätzen der Kreuzungsposition helfen. Nahe dem Startpunkt A beträgt er lediglich wenige Grad. Je mehr sich der Nutzer dem Punkt B nähert, desto größer wird der Differenzwinkel α . Befindet er sich mit B schließlich auf gleicher Höhe, beträgt $\alpha = 90^\circ$.

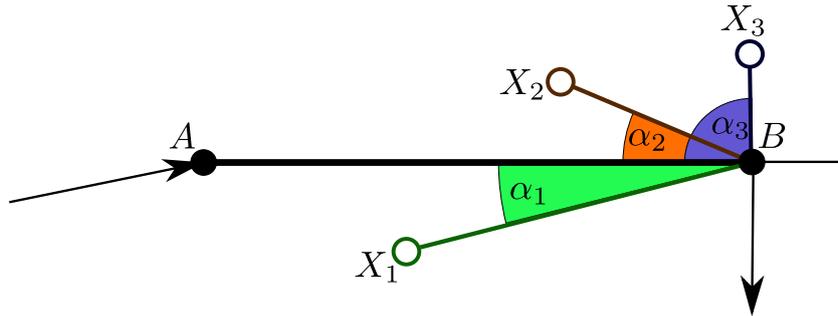


Abbildung 4.7.: Das Bild illustriert die Berechnung des Differenzwinkels α . Das Wegsegment verläuft vom Punkt A auf der linken Bildseite zu Punkt B rechts. Die Punkte X_1 , X_2 und X_3 stammen von der GPS Empfangseinheit. X_1 hat etwa $1/3$ von \overline{AB} zurückgelegt. Bei X_2 sind es circa $2/3$ des Weges. X_3 befindet sich schließlich auf gleicher Höhe mit dem Punkt B. Die Strecke \overline{XB} wurde zu Illustrationszwecken etwas verlängert.

Wird davon ausgegangen, dass der Punkt B genau die Kreuzungsmittle darstellt, so steht der Nutzer bereits auf der Straße, wenn α bei 90° liegt. Daher wird ein Schwellwert definiert, ab welchem der Punkt B als erreicht gilt. Wenn α diesen Schwellwert überschreitet, erfolgt die Ausgabe eines Warnhinweises in Verbindung mit der nächsten Routinganweisung. Der optimale Schwellwert wird im Abschnitt 6.1 durch einen Außentest bestimmt.

Straßenseitenerkennung

Des Weiteren wird der Versuch der Straßenseitenerkennung unternommen. Dabei soll ermittelt werden, ob der Nutzer eine Straße auf der linken oder rechten Seite entlangläuft. Ist die Straßenseite bekannt, können die Routinganweisungen präzisiert werden. Beispiele hierfür sind:

- Abbiegen an Kreuzungen: Der Nutzer wird zusätzlich zur Abbiegeanweisung darüber informiert, welche Straßen überquert werden müssen. Des Weiteren können Informationen über die Lage der Fußwege einbezogen werden.
- Ausgabe von selektiven Warnungen, wenn eine Kreuzung geradeaus überquert werden muss. Beispiel: Der Nutzer läuft die Hauptstraße entlang und auf der rechten Straßenseite mündet eine Nebenstraße (T-Kreuzung) ein. Befindet sich der Nutzer auf der rechten Straßenseite, so muss er vor der Nebenstraße gewarnt werden. Läuft er hingegen auf der linken Seite entlang, wird kein Hinweis benötigt.

Zur Veranschaulichung dient erneut Abbildung 4.7 auf Seite 41. Diesmal stellt sich jedoch die Frage, ob der Punkt X_n links oder rechts von \overline{AB} liegt. Daher wird auf die Bildung des Betrages verzichtet und der Winkel α wie folgt berechnet:

$$\alpha = \begin{cases} (R_{AB} - R_{XB}) + 360 & (R_{AB} - R_{XB}) < 180 \\ (R_{AB} - R_{XB}) & -180 \leq (R_{AB} - R_{XB}) \leq 179 \\ (R_{AB} - R_{XB}) - 360 & (R_{AB} - R_{XB}) > 179 \end{cases} \quad (4.5)$$

Wobei R_{AB} und R_{XB} wie gehabt die Himmelsrichtungen der Strecken in Grad darstellen und α zwischen -180 und 179 Grad liegt.

Ist der Winkel α positiv, so befindet sich der Nutzer auf der rechten Straßenseite, Bei einem negativen Wert läuft er auf hingegen auf der linken Seite.

4.6. Kreuzungsquerungen

Im Rahmen des durchgeführten Nutzertests (siehe Kapitel 6.2) wurden Schwierigkeiten beim Auffinden des optimalen Querungspunkts einer Kreuzung beobachtet. Die Probanden wurden anschließend nach ihrer Einschätzung zu dem Thema befragt. Zwei von drei bestätigten leichte Unsicherheiten bei der Querung von unbekanntem Kreuzungen. In diesem Abschnitt wird zunächst kurz auf die derzeitige Darstellung von Kreuzungen eingegangen, bevor schließlich ein erweitertes Kreuzungsmodell vorgestellt wird.

Momentan wird eine Kreuzung durch genau einen Punkt repräsentiert, welcher sich in der Mitte aller kreuzenden Straßen befindet. Dieser Punkt fungiert als nächstes Teilziel und wird mit Hilfe von Entfernungs- und Richtungsdaten anvisiert. An dieser Stelle muss erneut auf die relative Ungenauigkeit der GPS Daten hingewiesen werden, welche das Auffinden des Punktes erschwert. Die Ungenauigkeit ist allerdings nicht das Hauptproblem. In Abschnitt 4.5.4 wird ein möglicher Kompensationsmechanismus beschrieben.

4. Entwurf

Das Hauptproblem besteht hingegen darin, dass bisher keine Informationen über die optimalen Querungsstellen einer Kreuzung vorliegen. Somit kann lediglich die angesprochene Entfernung zur Kreuzungsmitte verwendet werden. Je breiter jedoch die Straßen, desto größer der Kreuzungsbereich und desto weiter ist die Querungsstelle vom Kreuzungsmittelpunkt entfernt. So kann es vorkommen, dass dem Nutzer noch eine Entfernung von 15-20 Meter mitgeteilt wird, er aber in Wahrheit bereits an einer der Straßen steht oder im Falle einer T-Kreuzung bereits an der vorgesehenen Querungsstelle vorbeigegangen ist. Letzteres passierte während des Nutzertests an der Kreuzung „Nöthnitzer Straße, Georg-Schumann Straße“.

Zur Lösung des Problems schlägt Peter Wendor ein Schema zur Modellierung von Querungsstellen vor[15, Kapitel 4.3]. Dafür muss zunächst die Darstellung einer Straße erweitert werden. Momentan besteht eine Straße aus einer Liste von Punkten. Zusätzliche Eigenschaften wie Fußwege oder Bodenbeschaffenheit werden dem Weg lediglich als Attribut-Tags hinzugefügt. In der erweiterten Darstellung bilden Straße und Bürgersteige separate Wege, welche in einer Relation gebündelt werden.

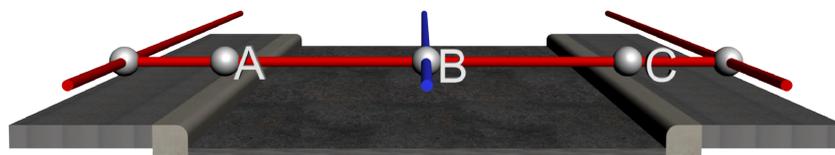


Abbildung 4.8.: Schema zur Abbildung von Querungen an Straßenkreuzungen. Eine Querung besteht aus 5 Punkten: Die Knoten links und rechts stellen die Verbindung zu den Bürgersteigen dar. A und C befinden sich auf Höhe der Bordsteinkante. B liegt auf der Mitte der Straße. Quelle: Abbildung 4.1 aus [15, Kapitel 4.3].

Eine Querungsstelle besteht nun aus fünf Punkten, die zu einem Weg zusammengefasst werden. Die äußeren Punkte bilden den Start- bzw. Endpunkt der Querung und stellen die Verbindung zu den Bürgersteigen her. Der mittlere Knoten (Punkt B) liegt auf der zu kreuzenden Straße. Die dazwischen liegenden Punkte A und C stellen Informationen über die Querung wie beispielsweise die Höhe der Bordsteinkante zur Verfügung (siehe Abbildung 4.8 auf Seite 43).

Wird dieses Modell für die Erfassung von Kreuzungen verwendet, muss nicht mehr der Kreuzungsmittelpunkt als nächster Routenpunkt dienen. Statt dessen wird der erste Knoten des Querungsstückes als Routenpunkt genutzt. Mit dem Erreichen des Punktes steht der Nutzer automatisch an der optimalen Querungsstelle. Im nächsten Schritt werden die Informationen über die Querung bereitgestellt.

Um das vorgestellte Modell einsetzen zu können, sind allerdings massive Änderungen an den Datenstrukturen der OpenStreetMap erforderlich. Dies betrifft vor allem die Repräsentation von Straßen als Relationen. Des Weiteren fehlen die Spuren für die Bürgersteige und die Daten für die Querungen. Beides müsste von der OSM Community nachgetragen werden. Dabei ist unklar, ob solch weitreichende Änderungen überhaupt

4. Entwurf

eine Mehrheit in der Community finden würde. Falls ja, würde dennoch viel Zeit vergehen, bis eine für die Routingzwecke ausreichende Menge an Daten gesammelt worden wäre.

Daher wird im Rahmen dieser Arbeit ein vereinfachtes Modell vorgeschlagen. Statt Bürgersteige einzeln abzulegen und Mehrpunkt-Querungsstellen zu speichern wird an jeder geeigneten Querung ein Knoten auf der Straße gesetzt. Dieser Knoten hält die Querungsinformationen wie die Höhe der Bordsteinkante oder das Vorhandensein von taktilen Leitlinien vor. Abbildung 4.9 auf Seite 44 veranschaulicht das Modell am Beispiel der „Nöthnitzer Straße, Regensburger Straße“. M stellt den bisher anvisierten Mittelpunkt der Kreuzung dar. Der als Querungspunkt beschriftete Knoten stellt eine geeignete Überquerungsmöglichkeit der Straße dar.



Abbildung 4.9.: Die Karte zeigt die Kreuzung Nöthnitzer Straße, Regensburger Straße. M ist der errechnete Kreuzungsmittelpunkt. Daneben ist eine geeignete Stelle zur Überquerung der Nöthnitzer Straße markiert. Diese befindet sich Nahe der Bushaltestelle Regensburger Straße. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

Auch dieses Modell kommt nicht ohne die Hilfe der OSM Community aus, denn die Querungspunkte müssen von den Mappern ebenfalls händisch eingefügt werden. Allerdings erfordert dies keine Umstrukturierung der gesamten Datenstrukturen. Tatsächlich wird ein Teil dessen bereits bei komplexen Ampelkreuzungen angewendet. Bei einer solchen Kreuzung wird für jede Ampel ein Knoten angelegt. Die Knoten liegen auf den einmündenden Straßen und kennzeichnen in den meisten Fällen auch den Querungspunkt für Fußgänger⁵. Diese Knoten müssen lediglich um die Querungsattribute ergänzt werden. Falls auf der betreffenden Straße keine Ampel existiert, wird der Knoten zuvor angelegt.

Um dieses Modell prototypisch umzusetzen wurden soweit vorhanden die Positionen der Ampeln und damit die potentiellen Querungsstellen dem Kreuzungsdatentyp hinzugefügt. Sie werden in der Kreuzungsdetailansicht nach Entfernung sortiert angezeigt (siehe 4.5.4).

⁵http://wiki.openstreetmap.org/wiki/Tag:highway%3Dtraffic_signals#Tag_all_incoming_ways abgerufen am 17.01.2014

4. Entwurf

Während des Routings werden sie allerdings noch nicht berücksichtigt. In wie fern diese Punkte lokalisiert werden können und falls ja, ob sie als Querungsstellen geeignet sind, müssen weitere Tests ergeben.

5. Implementierung

Nachfolgend werden die beschriebenen Konzepte des letzten Kapitels konkretisiert. Dabei werden zuerst die serverseitigen Komponenten und danach der Aufbau des Clients erläutert.

5.1. Server

Dieser Teil gliedert sich erneut in zwei Unterabschnitte. Zunächst erfolgt das Anlegen der OpenStreetMap Datenbank inklusive der für das Routing benötigten Erweiterungen. Im zweiten Teil wird der Aufbau der Komponenten zur Verarbeitung einer Routinganfrage beschrieben.

5.1.1. Installation und Konfiguration

Für die OSM Daten kommt das Datenbanksystem PostgreSQL¹ zum Einsatz. Zu Beginn wird PostgreSQL daher auf einem Testserver mit dem Debian Betriebssystem installiert.

Der Funktionsumfang von PostgreSQL kann durch Plugins erweitert werden. Dazu muss die betreffende Erweiterung zunächst auf dem Zielsystem installiert werden. In einem zweiten Schritt wird sie dann in die konkrete Datenbank geladen. Fortan stehen deren Funktionen wie beispielsweise zusätzliche Datentypen oder Funktionen bei der Arbeit mit der Datenbank zur Verfügung.

Eine der benötigten Erweiterungen ist PostGIS². PostGIS dient zur Speicherung und Verarbeitung von Geo-Daten wie Punkten, Linien und Flächen. Dadurch ist beispielsweise eine Abstandsberechnung zwischen zwei Punkten oder einem Punkt und einer Fläche möglich. PostgreSQL und PostGIS bilden zusammen eine Datenbank, die von Geoinformationssystemen verwendet werden kann.

Anlegen und Befüllen der Datenbank

Nach erfolgreicher Installation des Datenbankservers folgt das Anlegen einer neuen, leeren Datenbank namens „SachsenOSM“. Anschließend wird die PostGIS Erweiterung geladen. Die Datenbank ist nun für die OSM Geodaten vorbereitet.

Der nächste Schritt besteht im Import von geeignetem Kartenmaterial. Das OSM Projekt bietet die gesamte, wöchentlich aktualisierte Weltkarte zum Download an. Dessen Import benötigt allerdings sehr viel Zeit und Speicherplatz. Eine Alternative

¹<http://www.postgresql.org/> abgerufen am 08.01.2014

²<http://postgis.net/> abgerufen am 08.01.2014

bilden Kartenausschnitte, welche von einigen Firmen und Personen erstellt und angeboten werden. Karten von Deutschland und dessen Bundesländern sind bei der Geofabrik³ erhältlich. Im Rahmen dieser Arbeit kommt der Kartenausschnitt von Sachsen zum Einsatz. Nach dem Download wird er mit dem Programm Osmosis⁴ in die leere Datenbank importiert. Das gesamte Kartenmaterial steht unter der CC-BY-SA Lizenz⁵.

Neben den kompletten Karten werden auch minütliche, stündliche und tägliche Updates zum Download angeboten. Diese Dateien sind wesentlich kleiner und enthalten lediglich die Änderungen für das jeweilige Zeitintervall. Sie werden mit Hilfe von Osmosis in die lokale Datenbank eingepflegt. Um den Aktualisierungsprozess zu automatisieren, wurde ein Skript erstellt, welches die Kartenupdates herunterlädt und in die Datenbank integriert.

Kreuzungen

Dem auf diese Weise importierten Kartenmaterial fehlt allerdings eine Tabelle, welche Informationen über Kreuzungen bündelt. So müssen Informationen über die Kreuzungseigenschaften und deren Straßen aus unterschiedlichen Tabellen zusammengesucht werden. Dies beansprucht zu viel Zeit. Daher werden die Kreuzungsdaten in einer eigenen Datenstruktur neu gruppiert. Folgende Anforderungen müssen erfüllt werden:

1. Für eine gegebene Knoten-ID ist die Entscheidung zu treffen, ob es sich um eine Kreuzung handelt.
2. Falls ja, sollen Eigenschaften wie Name und Anzahl der Straßen sowie eine Liste aller Straßen-ID's abgefragt werden.

Die Informationen wurden auf zwei Tabellen aufgeteilt. Die Erste namens „Intersections“ besteht aus der Kreuzungsnode-ID, deren Namen und der Anzahl der Straßen. Jede Zeile entspricht somit einer Kreuzung. Eine zweite Tabelle namens „IntersectionData“ hält die ID's der einzelnen Straßen einer Kreuzung vor. Die Tabelle besteht aus den Spalten: Kreuzungsnode-ID, Weg-ID, Wegknoten-ID.

Dies lässt sich am Besten anhand eines Beispiels veranschaulichen. Angenommen es existieren die beiden Wege:

- Weg A mit ID 100 besteht aus den Knoten: 1, 2, 3, 4
- Weg B mit ID 200 besteht aus den Knoten: 2, 10, 11

Der Knoten mit der ID 2 ist eine T-Kreuzung, da er sowohl Weg A als auch Weg B angehört. Die „Intersections“ Tabelle 5.1 besitzt also die folgende Zeile:

In der Tabelle „IntersectionData“ (5.2) befinden sich hingegen 3 Zeilen, eine für jede mündende Straße:

³<http://www.geofabrik.de> abgerufen am 08.01.2014

⁴<http://wiki.openstreetmap.org/wiki/DE:Osmosis> abgerufen am 08.01.2014

⁵<http://creativecommons.org/licenses/by-sa/2.0/> abgerufen am 08.01.2014

5. Implementierung

Kreuzungsknoten-ID	Name	Anzahl Straßen
2	Weg A, Weg B	3

Tabelle 5.1.: Ausschnitt aus der Intersections Tabelle: Eintrag für Kreuzung Weg A, Weg B

Kreuzungsknoten-ID	Weg-ID	Wegknoten-ID
2	100	1
2	100	3
2	200	10

Tabelle 5.2.: IntersectionData Tabelle mit allen mündenden Straßen der Kreuzung Weg A, Weg B

Um zu ermitteln, ob ein gegebener Knoten eine Kreuzung ist, wird nach dessen ID in der „Intersections“ Tabelle gesucht. Fällt das Ergebnis der Suche positiv aus, werden mit einer zweiten Abfrage alle zugehörigen Straßen ermittelt. Die Weg-ID's liefern Informationen u.a. über den Straßennamen und -typ. Anhand der Wegknoten-ID's lassen sich ausgehend vom Kreuzungspunkt die Himmelsrichtungen der Straßen bestimmen.

Routing

Um den Routinggraphen aus Kapitel 4.5.2 zu erstellen, wird das Programm `osm2po`⁶ verwendet. `osm2po` erzeugt eine neue Tabelle in der „SachsenOSM“ Datenbank und füllt sie mit Wegsegmenten aus der Ways Tabelle. Es ist möglich, bestimmte Wegklassen herauszufiltern und sie somit nicht in das Routing einzubeziehen. Im Rahmen dieser Arbeit wurden Autobahnen und autobahnähnliche Straßen gefiltert, da sie für Fußgänger nicht zugänglich sind.

Die Tabelle 5.3 veranschaulicht das Ergebnis am Beispiel einiger Segmente der Nöthnitzer Straße in Dresden:

Source und Target sind die Knoten des Graphen und Length ist die Länge der Kante in km. Der Wert in der Class-Spalte ordnet das jeweilige Wegsegment in eine der Wegkategorien ein, die im Kapitel 4.5.2 aufgestellt wurden. Die Nöthnitzer Straße ist eine Tertierstraße und somit gehören alle dargestellten Segmente der Kategorie 1 an.

Im nächsten Absatz von Kapitel 4.5.2 werden die Kostenfunktion und die dafür benötigte Wichtungstabelle beschrieben. Letztere wird genau wie oben erläutert als eine neue Tabelle in der Datenbank angelegt.

Schließlich fehlen noch die Routing Algorithmen (Dijkstra und A*). Deren Implementation ist in der PostgreSQL Erweiterung `pgRouting`⁷ enthalten. Nach der Installation und

⁶<http://osm2po.de> abgerufen am 08.01.2014

⁷<http://pgrouting.org> abgerufen am 09.01.2014

ID	Source	Target	Length	Class
8778	173208	173197	0.010139427	1
8779	173197	173199	0.00974001	1
8780	173199	173205	0.013539517	1
8781	173205	613653	0.010798447	1
8782	613653	173201	0.024122909	1

Tabelle 5.3.: Ausschnitt aus der von osm2po erstellten Routengraphen Tabelle. Die fünf Zeilen sind Teilstücke der Nöthnitzer Straße in Dresden.

dem Laden in die „SachsenOSM“ Datenbank ist der Einrichtungsprozess abgeschlossen und die Datenbank kann für das Routing eingesetzt werden.

5.1.2. Aufbau der Komponenten

Dieser Absatz beschreibt den Aufbau der Serverkomponenten. Die Entwicklung erfolgte in der Programmiersprache Python.

Für die Kommunikation mit dem Client wird ein Webserver benötigt. Dieser wird mit dem Web Framework CherryPy⁸ erstellt. Der Webserver ist über eine Domain erreichbar und nimmt Anfragen auf einem spezifischen Port entgegen. Die folgenden Funktionen wurden implementiert:

- GetRoute: Berechnet für einen Start- und Zielpunkt eine Route
- GetPOI: Listet alle POI einer Kategorie im Umkreis eines Koordinatenpaares auf
- GetDepartures: Gibt die nächsten Abfahrten für eine Haltestelle zurück

Sowohl die Anfragen als auch die Ausgaben erfolgen mittels JSON⁹. Das JSON Datenformat besteht aus Listen und Schlüssel-Wert Paaren und wird von vielen Programmiersprachen nativ gelesen und geschrieben. Daher bietet es sich für den Datenaustausch zwischen Client und Server an.

Nachdem die eingegangene Routinganfrage auf Vollständigkeit und Korrektheit überprüft wurde, erfolgt der Aufruf der RouteCreator Klasse. Diese ist für das Zusammen setzen der Route verantwortlich. Die eigentliche Berechnung der Routenteile erfolgt getrennt nach Fußweg und öffentlichen Verkehrsmitteln in zwei Subklassen namens FootwayCreator und TransportCreator.

Wenn die Route lediglich zu Fuß zurückgelegt werden soll, so reicht der RouteCreator die Anfrage nur an den FootwayCreator weiter und verfährt umgekehrt genauso mit dessen Resultaten. Aufbau und Funktion der FootwayCreator Klasse wurden bereits im Kapitel 4.5.2 skizziert und werden an dieser Stelle nicht weiter ausgeführt.

⁸<http://www.cherrypy.org> abgerufen am 10.01.2014

⁹<http://www.json.org/> abgerufen am 12.01.2014

5. Implementierung

Bei der Einbeziehung des ÖPNV wird von der TransportCreator Klasse hingegen zunächst die beste Verbindung zwischen Start und Ziel ermittelt (siehe Kapitel 4.5.3). Anschließend erfolgt die Berechnung der zu Fuß zurückzulegenden Route und schließlich die Verknüpfung aller Teilstücke.

Ferner existieren die folgenden Klassen:

- DBControl: Die Klasse stellt einen Wrapper für die Datenbankabfragen zur Verfügung. Sie nimmt SQL Queries entgegen und gibt deren Ergebnisse zurück.
- Geometry: Beinhaltet Funktionen zur Berechnung von Entfernung und Richtung zwischen zwei Geo-Koordinaten.
- POI: Erwartet ein Paar Geo-Koordinaten, den Umkreis in Meter und eine Liste von Kategorien und gibt eine Liste der zutreffenden POI zurück.
- ObjectParser: Erzeugt Wegpunkt-, Kreuzungs- und POIObjekte anhand einer OSM Node ID. Eine Auflistung von deren Attributen findet sich im Kapitel 4.1. Die Funktionen dieser Klasse werden u.a. von der obigen POI Klasse benötigt.
- Translator: Übersetzt Bestandteile der Route in verschiedene Sprachen. Momentan werden Deutsch und Englisch unterstützt. Die Sprache teilt der Client innerhalb der Routinganfrage mit.
- Config: Hält alle Konfigurationsdaten des Programms vor.

Die Rohdaten für die ÖPNV Verbindungen stammen von der Deutschen Bahn. Da die Bahn keine eigene API für den Abruf der Daten anbietet, kommt statt dessen die Java Bibliothek Public-Transport-Enabler¹⁰ zum Einsatz. Die unter der GPL stehende Bibliothek stellt eine Schnittstelle zur Abfrage der nächsten Verbindungen von einer Start- zur Zielhaltestelle bereit. Um die Java Library innerhalb des Python Programmcodes verwenden zu können, wird Py4J¹¹ verwendet. Py4J stellt eine Brücke zwischen Python und Java dar. Innerhalb des Python Codes ist der Zugriff auf Java Objekte inklusive ihrer Methoden somit nativ möglich.

¹⁰<http://code.google.com/p/public-transport-enabler/> abgerufen am 12.01.2014

¹¹<http://py4j.sourceforge.net> abgerufen am 12.01.2014

5.2. Client

Der Client wurde in der Programmiersprache Java für das mobile Betriebssystem Android ab Version 4.1 entwickelt. Der Code liegt innerhalb eines Eclipse Projekts vor. In den folgenden Absätzen wird der Programmaufbau näher beschrieben.

Grundsätzlich lässt sich das Programm in drei Teile zerlegen:

1. Routenobjekte
2. Utilities
3. Benutzungsoberfläche

Routenobjekte

Die im Folgenden beschriebenen Objektklassen implementieren die Routenobjekte, welche im Kapitel 4.1 entworfen wurden:

- Point: Diese Klasse stellt den Wegpunkt-Basistyp dar.
 - POIPoint: Erbt von Point und erweitert die Basisklasse um die POI Attribute
 - * StationPoint: Erbt von POIPoint und beinhaltet zusätzlich Daten über abfahrende Linien.
 - IntersectionPoint: Erbt ebenfalls von Point und enthält außerdem eine Liste von Kreuzungspunkten.
- FootwaySegment: Enthält Informationen wie Name und Länge über ein zu Fuß zurückzulegendes Wegstück.
- TransportSegment: Enthält Informationen wie Abfahrts- und Ankunftszeit sowie Linie und Fahrtziel über einen mit öffentlichen Verkehrsmitteln zurückzulegenden Routenabschnitt.

Neben den Get- und Setmethoden für die einzelnen Attribute implementiert die Point Basisklasse auch die folgenden Funktionen:

- `int distanceTo(Point p)`: Berechnet die Entfernung des aktuellen Objekts zu einem gegebenen Punkt p in Metern.
- `int bearingTo(Point p)`: Errechnet die Richtung zu einem gegebenen Punkt p in Grad. Dies entspricht der Himmelsrichtung des entstehenden Segments.
- `int hashCode()`: Bildet anhand der Breitengrad- und Längengrad Werte einen individuellen Hashwert.
- `boolean equals(Object obj)`: Überprüft anhand von Längen- und Breitengrad, ob zwei Point Objekte identisch sind.

5. Implementierung

- `JSONObject toJson()`: Serialisiert das Point Objekt, indem es in das Json Datenformat überführt wird.
- `String toString()`: Gibt eine Beschreibung des Routenobjekts zurück.

Durch die Vererbung stehen diese Funktionen auch für die aufbauenden Point-Klassen zur Verfügung. Die `toJson()` und `toString()` Funktionen werden von den Klassen um die jeweilig hinzukommenden Variablen erweitert.

Zusätzlich existiert eine Wrapper-Klasse, welche als eine Art Oberklasse aller Routenobjekte fungiert. Eine Instanz dieser Klasse hält genau ein Routenobjekt vor. Damit werden Point- und Segmentklassen gruppiert und es entsteht ein nach außen einheitliches Routenobjekt.

Utilities

Dieser Abschnitt befasst sich mit den Utility Klassen:

- `PositionManager`: Dient zur Erfassung und Verarbeitung der eingehenden Positionsdaten. Quellen sind sowohl der GPS Empfänger als auch der Netzwerkprovider (A-GPS). In der Funktion `boolean isBetterLocation(Location location, Location currentBestLocation)` wird entschieden, ob eine neue Position akzeptiert oder verworfen wird.
- `SensorsManager`: Die Klasse stellt Daten über sonstige Sensoren des Smartphones bereit. Momentan werden Kompass und Beschleunigungssensor abgefragt.
- `DataDownloader`: Versendet Anfragen an einen Server und verarbeitet dessen Antworten. Es werden sowohl die GET als auch die POST Methode unterstützt.
- `SettingsManager`: Lädt und speichert Programmeinstellungen und weitere anfallende Daten wie die zuletzt besuchten Orte und angelegte POI Kategoriepresets. Die Daten werden in das Json Datenformat übertragen und im internen Speicher des Smartphones abgelegt. Des Weiteren stellt die Klasse Methoden zur Abfrage und Änderung der Daten bereit.
- `ObjectParser`: Die Funktionen dieser Klasse nehmen einen Json kodierten Datenstrom entgegen und geben nach dessen Verarbeitung eines der Routenobjekte zurück. Für jedes der sechs Routingobjekte existiert eine eigene Methode.
- `Globals`: Diese statische Klasse ist von überall her aufrufbar und fungiert als temporärer Speicher. Außerdem hält sie Objektinstanzen der Klassen `PositionManager`, `SensorsManager` und `SettingsManager` vor und implementiert so das Singleton Pattern. Grund dafür ist, dass alle drei Klassen nicht mehrfach instanziiert werden dürfen denn es existiert beispielsweise nur ein GPS Empfänger.

Die Klassen `DataDownloader`, `PositionManager` und `SensorsManager` stellen allesamt Informationen für andere Programmteile zur Verfügung. Das `DataDownloader` Objekt informiert über die Antwort auf eine Serveranfrage und leitet die Antwortdaten weiter. `PositionManager` und `SensorsManager` stellen aktualisierte Sensordaten bereit. Zur Verteilung der Daten existiert in jeder Klasse ein Interface, welches von den Abnehmerklassen implementiert werden muss, um an die Daten zu gelangen.

Daneben existiert die `Route` Klasse zur Repräsentation einer abgerufenen Route. Die Klasse besitzt Funktionen zum Hinzufügen von Routenobjekten, zur Ausgabe des aktuellen Objektes sowie zur Wahl des vorherigen bzw. nächsten Routenpunktes. Intern werden die einzelnen Routenobjekte in einer `ArrayList` gespeichert. Die Liste beherbergt die Objekte allerdings nicht direkt, sondern verwaltet stattdessen die oben beschriebenen `RouteObjectWrapper` Instanzen.

Benutzungsoberfläche

Die Benutzungsoberfläche eines Android Programms besteht aus einer oder mehreren `Activities`. Eine `Activity` bildet die Basis für weitere Oberflächenobjekte und dient zur Kommunikation mit dem Nutzer. Oberflächenkomponenten wie `Buttons`, `Textfelder` oder `Eingabefelder` werden innerhalb eines `View Container` dargestellt. Alternativ kann eine `Activity` auch zunächst aus mehreren `Fragments` bestehen, welche dann ihrerseits einen `View Container` für die `User Interface` Komponenten besitzen.

`Activities` und `Fragments` implementieren u.a. die folgenden Methoden:

- `OnCreate`: Wird beim ersten Aufruf des Objekts ausgeführt. Hier werden Variablen initialisiert und die Komponenten der Benutzungsoberfläche angelegt und implementiert.
- `OnPause`: Wird aufgerufen, wenn die `Activity` in den Hintergrund verlagert wird, z.B. wenn eine andere `Activity` geladen wird.
- `OnResume`: Bildet das Gegenstück zu `OnPause`.

Das Layout der Benutzungsoberfläche wird entweder statisch innerhalb einer `XML` Datei oder dynamisch innerhalb der `OnCreate` Methode definiert.

Beim Start des Clients wird zunächst die `MainActivity` geladen. Diese besteht wiederum aus drei `Fragments`, wobei immer nur eines davon dargestellt wird. Die `Activity` stellt die Menüleiste und ist für die Organisation der `Fragments` verantwortlich. Die einzelnen `Fragments` stellen jeweils eine Benutzungsoberfläche bereit und bilden zusammen den Hauptteil der Anwendung.

Im Start `Fragment` werden Start- und Zielpunkte sowie Optionen der zu planenden Route festgelegt. Es wird unmittelbar nach dem Laden der `MainActivity` angezeigt und bildet den Einstiegspunkt für den Nutzer. Nach der Auswahl des Start- und Zielpunktes erfolgt die `Json` formatierte Routenanfrage an den Server, welcher in den Programmeinstellungen hinterlegt ist. Um dessen Antwort empfangen und verarbeiten zu können, wird das Interface namens `DataDownloadListener` implementiert, das die `DataDownloader`

5. Implementierung

Klasse bereitstellt. Fällt die Antwort des Servers positiv aus, wird aus den erhaltenen Daten unter Zuhilfenahme der `ObjectParser` Klasse eine Route erstellt und dem Router Fragment übergeben. Anderenfalls wird der Nutzer über den aufgetretenen Fehler benachrichtigt.

Der Router Tab besteht aus zwei Subfragmenten. Das Eine beinhaltet lediglich eine Liste, welche einen Überblick über die Route gibt. Das andere Subfragment weist ein komplexeres Layout auf und dient der Darstellung des nächsten Routenpunktes.

Das dritte Fragment wird für die Anzeige der Points of Interest in der Umgebung benötigt. Es besteht vornehmlich aus einer Combobox zur Auswahl der POI-Kategorie, einem Eingabefeld zur Suche und einer Liste, welche die gefundenen POI darstellt. In einer separaten Activity namens `POIPresetsActivity` lassen sich neue POI Kategorien anlegen oder bestehende bearbeiten.

Alle drei Fragments implementieren die bereitgestellten Interfaces der `SensorsManager` und `PositionManager` Klassen um die aktuellen Positions- und Kompassdaten zu erhalten. Des Weiteren besitzen alle drei Fragments eine Funktion, die deren Benutzungsoberflächen aktualisiert. Auf diese Weise kann beispielsweise auf neue Positionsdaten reagiert werden.

Neben diesen Hauptkomponenten existieren noch ein paar weitere Activities:

- Die `RouteDetailsActivity` stellt alle Optionen eines ausgewählten Routenobjekts dar und bietet die Möglichkeit, selbiges als Start oder Ziel einer Route auszuwählen (siehe 4.5.4).
- Unter der `GPSStatusActivity` lassen sich die Eigenschaften des GPS Signals einsehen.
- Die `EnterAddressActivity` dient zur Eingabe von Start- oder Zieladresse.
- Die `HistoryActivity` listet die zuletzt besuchten Punkte auf.
- Die `SettingsActivity` bietet die Möglichkeit, die Programmeinstellungen zu bearbeiten.

6. Evaluation

Dieses Kapitel beschäftigt sich mit der Evaluation des implementierten Prototypen. Zuerst folgt eine Reihe von Funktionstests. Anschließend wird ein Nutzertest durchgeführt. Schließlich wird die Bewertungsfunktion für ÖPNV Verbindungen evaluiert.

6.1. Funktionstest

Bevor der Nutzertest durchgeführt werden kann, muss zunächst sichergestellt werden, dass der Client für die Gruppe der blinden und sehbehinderten Probanden benutzbar ist (Anforderungsgruppe A3). Ferner wird überprüft, ob den vom Server errechneten Routen vom Start- zum Zielpunkt gefolgt werden kann (A1.1) und ob die empfangenen Positionsdaten den Genauigkeitsvorgaben genügen (A1.2). Schließlich werden die Entwürfe aus dem Kapitel 4.5.4 auf ihren Nutzen hin überprüft. Die jeweiligen Tests wurden vom Autor der Arbeit als Expertentests durchgeführt.

Client

Als Testgerät kommt ein Samsung Nexus S mit der Android Version 4.1.2 zum Einsatz. Neben dem 1GHz Prozessor, 512MB Arbeitsspeicher und einem GSM Modul besitzt das Smartphone einen 4 Zoll Bildschirm, GPS Empfänger, sowie Kompass und Beschleunigungssensor.

Zuerst wurde überprüft, ob die Hardware des Clients der Anforderung A3.1 genügt. Das Gerät ist mit den Abmessungen von 123,9x63,0x10,8mm (HxBxT) problemlos in einer Hand zu halten. Die mobile Internetnutzung funktioniert nach dem Einlegen einer SIM Karte ebenfalls problemlos.

Das GPS-Empfangsmodul liefert sekundlich aktualisierbare Positionsdaten. Die Genauigkeit liegt in den meisten Fällen bei 10-20 Metern. Zusammen mit den beschriebenen Filterstrategien aus Kapitel 4.5.4 genügt dies für eine stabile Navigation. Dennoch wäre eine exaktere Positionsbestimmung selbstverständlich wünschenswert.

Kompass, Beschleunigungsmesser und die Vibrationseinheit arbeiten ebenfalls zufriedenstellend. Der Kompass zeigt die Himmelsrichtung außerhalb von Gebäuden bis auf wenige Grad genau an. Allerdings muss er nach dem Neustart des Smartphones gelegentlich kalibriert werden. Zu erkennen ist dies, wenn sich die Kompasswerte bei einer vollen Drehung nur noch um wenige Grad ändern oder der Kompass dauerhaft null Grad zeigt. Um dies schnell und einfach erkennen zu können, wurde die Routenverfolgungs-Darstellung um den Kompasswert ergänzt.

Das Testgerät erfüllt somit alle Punkte der Anforderung A3.1. Außerdem wurde noch die maximale Nutzungsdauer mit einer Akkuladung ermittelt. Sie liegt bei etwa 2,5 Stunden. Diese relativ kurze Zeitdauer kommt durch das hohe Aktualisierungsintervall der GPS Daten zustande. Je öfter eine aktualisierte GPS Position abgefragt wird, desto genauer kann der Standort bestimmt werden aber desto höher liegt auch der Stromverbrauch.

Negativ fiel auf, dass die Bedienung des Smartphones bei Regen nur noch sehr eingeschränkt möglich ist. Die Gesten zur Steuerung werden auf einem nassen Display nicht mehr erkannt.

Anforderung A3.2 verlangt nach kontinuierlichem akustischen Feedback während der Bedienung. Dies wird durch den Einsatz des integrierten Screenreaders Talkback sichergestellt. Während der Tests kam Version 3.5.0 zum Einsatz. Ist Talkback aktiviert, wird bei Berührung eines Objekts dessen Name vorgelesen. Für Sehbehinderte bietet das Android Betriebssystem die Möglichkeit der Schriftgrößenanpassung.

Bei der Oberflächengestaltung des Prototypen kamen ausschließlich Standard Komponenten wie Textfelder, Buttons und Listen zum Einsatz. Eine dauerhaft sichtbare Menüleiste am oberen Bildschirmrand garantiert den einfachen Wechsel zwischen den Programmfunktionen. Das User Interface wurde auf die folgenden Kriterien hin überprüft[11]:

- Alle Steuerelemente besitzen eine aussagekräftige Beschreibung.
- Bei Eingabefeldern ist ein Hinweistext vorhanden, welcher Auskunft über den einzutragenden Inhalt gibt.
- Alle Oberflächenobjekte sind sowohl durch die Funktion „Explore by Touch“ wie auch durch Navigationsgesten erreichbar. Die Fokusreihenfolge folgt dem Programmaufbau.

Die Einhaltung dieser Punkte garantiert die Benutzbarkeit der Programmoberfläche durch Screenreader Nutzer (Anforderung A3.3).

Das in A3.4 geforderte Vibrationsfeedback wird ebenfalls durch Talkback bereitgestellt. Einige selbst definierte Vibrationspattern werden zusätzlich verwendet, z.B. um das Ende einer Ladephase oder das Erreichen eines Routenpunktes zu signalisieren.

Damit werden alle Anforderungen aus A3 erfüllt und die Tauglichkeit für die Zielgruppe der blinden und sehbehinderten Nutzer bestätigt.

Fußgängerrouen

Um die Eignung der erstellten Fußgängerrouen beurteilen zu können, wurden vier verschiedene Rouen im Raum Dresden geplant und abgelaufen. Die Strecken verliefen sowohl durch die Innenstadt als auch durch Gebiete am Stadtrand. Vorrangiges Ziel war die Überprüfung der errechneten Route auf Konsistenz und Vollständigkeit. Ferner sollte auf diese Weise ermittelt werden, ob die empfangenen GPS Daten eine ausreichende Genauigkeit aufweisen, um für die Navigation eingesetzt werden zu können.

Die folgenden Rouen wurden abgelaufen:

6. Evaluation

1. Straßenbahnhaltestelle Pulsnitzer Straße -- Restaurant Saponi d'Italia (Rothenburger Straße 46 / Ecke Louisenstraße)
2. Fakultät Informatik (Nöthnitzer Straße 46) -- Cämmerswalder Straße 1
3. Cämmerswalder Straße 1 -- Rathaus Plauen (Nöthnitzer Straße 2)
4. Firat Kebap-Haus (Bergstraße 68) -- Eisenstückstraße 3

Die genauen Routenverläufe finden sich im Anhang A.2.

Alle erstellten Routen konnten erfolgreich abgelaufen werden. Sämtliche auf der Strecke liegende Kreuzungen waren in der Routenbeschreibung vorhanden. Die Navigationsanweisungen erwiesen sich zusammen mit den Informationen über den Aufbau einer Kreuzung als ausreichend (A1.1).

Allerdings lässt sich dies nicht uneingeschränkt auf alle Kreuzungen übertragen. Ab einer gewissen Größe und Komplexität besteht eine Kreuzung nicht mehr aus einem einzelnen Punkt sondern umfasst ein Areal von Straßen und Wegen. Ein Beispiel für einen solchen Kreuzungsbereich ist der Dresdner Albertplatz. Diesen anhand einer erstellten Route ohne ein gewisses Maß an Ortskenntnissen zu überqueren, wird nicht für jeden Nutzer möglich sein.

Die GPS Daten konnten sowohl zu Fuß als auch innerhalb öffentlicher Verkehrsmittel kontinuierlich empfangen und ausgewertet werden. Deren Genauigkeit lag draußen bei etwa 10 Meter, in Fahrzeugen auf Grund der schlechteren Empfangsmöglichkeiten zwischen 20 und 25 Meter. Die Filterstrategien aus Kapitel 4.5.4 verhinderten größere Schwankungen des Signals. Dennoch wurden die Anforderungen aus A1.2 nur teilweise erfüllt. Das GPS Signal lag zwar kontinuierlich vor und die Warnungen bei dessen drastischer Verschlechterung wurden auch dargestellt, jedoch ist die geforderte Genauigkeit von einem Meter nicht erreicht worden.

Kreuzungspunkterkennung

In diesem Abschnitt wird das Konzept zur präziseren Kreuzungserkennung (unterer Teil von Kapitel 4.5.4) erprobt. Ziel ist neben der generellen Tauglichkeitsbeurteilung die Festlegung eines Schwellwertes, ab dem die Kreuzung als erreicht gilt.

Für die Benachrichtigung beim Erreichen der Kreuzung wird ein Schwellwert für den Winkel α gesucht. Ist α größer als der Schwellwert S , so gilt die Kreuzung als erreicht. Bereits während der Entwurfsphase wurde geklärt, dass S kleiner als 90° sein muss, da der Nutzer bereits auf der kreuzenden Straße steht, wenn $\alpha = 90^\circ$. Während des oben beschriebenen Tests der Routen wurde gleichzeitig mit verschiedenen Schwellwerten experimentiert.

Nach den ersten Versuchen stellte sich heraus, dass der optimale Schwellwert von der Größe der Kreuzung und damit von der Breite der Wege abhängt. Bei einer kleinen Kreuzung z.B. von zwei Fuß- oder Feldwegen wird die Abzweigung in dem Moment erreicht, wenn der Nutzer direkt auf dem Kreuzungspunkt anlangt und sich somit in der Mitte der Kreuzung befindet. Die Mitteilung über das Erreichen der Abzweigung soll

allerdings 1-2 Meter vorher erfolgen. Daraus ergibt sich ein Schwellwert etwas unterhalb von 90° . Nach einer weiteren Testreihe wurde der Schwellwert S_1 auf 70° festgelegt.

Für große Kreuzungen wird ein geringerer Schwellwert benötigt. Eine Kreuzung gilt als groß, wenn mindestens eine der Straßen breiter als 5 Meter ist. Da die Breite der Straßen allerdings selten bekannt ist, wird stattdessen auf den Straßentyp ausgewichen. Als große Straßentypen gelten Wohngebietsstraßen, Spielstraßen, Tertiär-, Sekundär- und Primärstraßen. Für S_2 wurde ein Wert von 40° ermittelt.

Beide Schwellwerte wurden so gewählt, dass sie bei einer maximalen Anzahl von Kreuzungen optimale Werte liefern. Anders formuliert stellen sie einen Kompromiss dar, welcher möglichst viele Kreuzungen abdeckt. Das bedeutet allerdings, dass die Benachrichtigungen nicht bei jeder Kreuzung exakt zur richtigen Zeit erscheinen. Um den optimalen Schwellwert für jede Kreuzung separat berechnen zu können, fehlen Informationen über die Straßenbreite sowie geeignete Querungspunkte.

Die Benachrichtigung gilt als erfolgreich, wenn sie maximal 4 Meter vor dem eigentlichen Kreuzungspunkt angezeigt wird. Für kleine Kreuzungen wurde eine hohe Treffergenauigkeit ermittelt. Diese liegt bei 90% d.h. 9 von 10 Kreuzungen wurden korrekt angekündigt. Große Kreuzungen hingegen weisen lediglich einen Genauigkeitswert von ca. 75% auf, wobei die Benachrichtigung bei den restlichen 25% etwa zur Hälfte zu früh oder zu spät ankam.

Für kleine Kreuzungen bieten die zusätzlichen Benachrichtigungen daher einen wirklichen Mehrwert. Dies gilt vor allem dann, wenn mehrere Wege kurz nacheinander abzweigen und der korrekte Weg mit dem GPS Signal nicht eindeutig identifiziert werden kann. Bei großen Straßen sollte sich der Nutzer hingegen nicht ausschließlich auf diese Hinweise verlassen.

Straßenseitenerkennung

Schließlich folgt der Test des Konzepts zur Straßenseitenerkennung aus Kapitel 4.5.4.

Zur Evaluation der Straßenseitenerkennung wurde die aktuell erkannte Straßenseite während der Routenverfolgung hinter dem Kompasswert dargestellt. Sie wurden beim Ablaufen mit getestet, indem zwischendurch gelegentlich die Straßenseite gewechselt wurde. Dabei gilt: Ist α negativ, so befindet sich der Nutzer links von \overline{AB} und somit auf der linken Straßenseite. Bei positiven Werten von α ist es dementsprechend die rechte Seite der Straße.

Auch an dieser Stelle hängen die Testergebnisse von der Breite der Straße ab. Je breiter eine Straße, desto sicherer wird die Straßenseite erkannt da der Abstand vom Rand zur idealisierten Straßenmitte zunimmt. Des Weiteren begünstigt eine breitere Straße zumeist den GPS Empfang.

Diese Erkenntnisse werden von den Testergebnissen widergespiegelt. Bei einer breiten Hauptverkehrsstraße funktionierte die Erkennung der Straßenseite in den meisten Fällen. Je schmaler die Straßen, desto weniger Aussagekraft kann den Ergebnissen zugeschrieben werden. Probleme treten auch bei eng bebauten Straßen auf, bei denen sich unmittelbar links und rechts neben den Bürgersteigen mehrstöckige Gebäude befinden.

Fazit: Bei der derzeit verfügbaren GPS Genauigkeit kann die Straßenseite nicht zuverlässig erfasst werden. Dafür liefert sie in zu vielen Situationen ungenügende Ergebnisse. Verbesserungen sind allerdings bei einer signifikanten Verbesserung der Positionsgenauigkeit zu erwarten.

6.2. Nutzertest

Im vorherigen Kapitel wurde vom Autor des Dokuments im Rahmen eines Expertentests die prinzipielle Tauglichkeit von Client und Routen nachgewiesen. Diese Ergebnisse werden in den folgenden Abschnitten im Rahmen eines Nutzertests überprüft.

Ferner soll ermittelt werden, ob die Tester in dem zum Einsatz kommenden Prototypen einen Mehrwert zu ihren bisherigen Navigationstechniken sehen. Dazu werden sie im Vorhinein über ihre Strategien zum Auffinden eines unbekanntes Ortes befragt. Hinterher werden die Tester um einen Vergleich ihrer Methoden mit dem Prototypen gebeten.

Der Nutzertest gliedert sich in die folgenden Teilabschnitte:

- Zuerst wird eine Einführung in das Projekt und den Aufbau des Prototypen gegeben. Anschließend werden dem Probanden einige Fragen gestellt. Dies geschieht im Rahmen eines Interviews auf Grundlage eines vorbereiteten Fragebogens.
- Danach erfolgt der Test, bei welchem zwei gestellte Aufgaben zu lösen sind.
- Zum Schluss wird ein weiteres Interview durchgeführt, welches der Auswertung der gelösten Aufgaben dient.

Der vollständige Fragebogen befindet sich im Anhang A.3.

Teil 1: Vorbefragung

An der Evaluation nahmen drei Personen teil. Zwei davon sind zwischen 20 und 30 Jahre, die Dritte zwischen 30 und 40 Jahre alt. Zwei Probanden sind Studenten, der Dritte hat sein Studium bereits beendet. Alle Teilnehmer sind blind. Sofern ein Restsehvermögen vorhanden ist, wirkt es sich nicht auf das Orientierungsvermögen aus. Weitere körperliche Einschränkungen lagen bei keinem der Probanden vor.

Bei der Einschätzung der generellen Orientierungsfähigkeiten in fremden Umgebungen gaben Zwei an, dass sie sich gut orientieren können. Die übrige Testperson schätzte ihre Orientierungsfähigkeiten auf einer Skala von 1 (sehr gut) bis 5 (sehr schlecht) als sehr gut ein.

Alle drei Probanden besitzen ein Handy oder Smartphone. Die folgenden Modelle wurden genannt:

- Nokia E66 (Symbian), Screenreader: Talks, Nutzung: 4 Jahren
- Apple iPhone 4s (iOS), Screenreader: VoiceOver, Nutzung: 2 Jahre
- Samsung S+ (Android), Screenreader: Talkback, Nutzung: 2 Jahren

Alle benutzen die Geräte täglich. Die Frage nach der Einschätzung des Erfahrungsgrades beantworteten zwei mit hoch und einer mit mittel.

Im nächsten Abschnitt wurde nach Strategien zum Auffinden eines unbekanntes Ortes gefragt. Alle drei Tester erkundigen sich im Voraus bei Freunden und Bekannten nach einer Route. Einer davon benutzt gelegentlich auch Google Maps, um sich eine Route, bestehend aus Textanweisungen erstellen zu lassen. Bei weiteren Strecken suchen alle Teilnehmer auch nach einer geeigneten ÖPNV Verbindung, um die zu Fuß zurückzulegende Strecke zu minimieren.

Unterwegs kommt bei allen Teilnehmern der weiße Langstock als Basishilfsmittel zum Einsatz. Die vorbereitete Route wird entweder aus dem Gedächtnis oder anhand getätigter Notizen abgelaufen. Bei Problemen erbitten alle die Hilfe von Passanten. Einer beschreibt ferner ein Live-Routing, bei dem ein sehender Bekannter die nächsten Routinganweisungen unter Zuhilfenahme von Google Maps per Telefon durchgibt.

Konkret auf Erfahrungen mit Navigationsprogrammen angesprochen, fallen die Antworten unterschiedlich aus. Einer verwendet gelegentlich Nokia Maps¹ die anderen zwei haben noch keine Erfahrung mit Navigationsprogrammen. Ein Proband hat bisher noch keine Programme gefunden, die ihm einen Mehrwert bei der Orientierung und Navigation bringen würden. Ein Anderer hatte sich bisher noch gar nicht mit den existierenden Lösungen auseinandergesetzt.

Teil 2: Navigationsaufgabe

Nach der Einführung in den Aufbau des Prototypen und der Vorbefragung der Nutzer, wird in diesem Abschnitt auf die Lösung der gestellten Aufgabe eingegangen.

Die Aufgabe lautet: Begib dich mit Hilfe des Navigationsprototypen zum Haupteingang des Dresdner Hauptbahnhofes. Die Aufgabe gliedert sich in zwei Teilaufgaben:

1. Wähle den Start- und Zielpunkt. Als Startpunkt dient die aktuelle Position. Dies ist die Fakultät Informatik der TU Dresden. Zielpunkt ist der bereits erwähnte Haupteingang des Hauptbahnhofes. Anschließend ist eine Route unter der Einbeziehung des ÖPNV zu erstellen.
2. Folge der Route aus Teilaufgabe 1.

Die Probanden wurden gebeten, während der Lösung der Aufgaben laut zu denken. Auf diesem Weg können mögliche Unklarheiten schnell aufgeklärt werden. Der Testleiter verhält sich weitestgehend passiv. Er beantwortet jedoch aufkommende Fragen und gibt gegebenenfalls auch Tipps zum weiteren Vorgehen.

Die prototypische Clientsoftware läuft auf einem Samsung Nexus S Smartphone mit der Android Version 4.1.2. Die erste Teilaufgabe wurde noch innerhalb der Fakultät Informatik bearbeitet. Alle drei Probanden waren in der Lage, Start und Ziel festzulegen. Zwei Teilnehmer wünschten sich allerdings eine detailliertere Schritt für Schritt Anleitung für die Auswahl eines POI als Zielpunkt.

¹<http://nokia-maps.softonic.de/symbian> abgerufen am 15.01.2014

6. Evaluation

Anschließend wurde die Route berechnet. Dies nahm etwa 30 Sekunden in Anspruch und wurde von allen Probanden als ausreichend schnell beurteilt. Zunächst wurde den Testern Zeit gegeben, um sich mit der erstellten Route in der Übersichtsansicht vertraut zu machen. Die Testroute gliederte sich grob in die folgenden Anweisungen:

1. 100 Meter auf der Nöthnitzer Straße, danach nach rechts in die Georg-Schumann Straße abbiegen
2. Der Georg-Schumann Straße 300 Meter folgen und die Kreuzung Georg-Schumann Straße, Münchner Platz geradeaus überqueren
3. 50 Meter geradeaus, anschließend nach rechts in die Münchner Straße abbiegen
4. 60 Meter bis zur Straßenbahnhaltestelle Münchner Platz
5. Fahrt mit der Linie 3 nach Dresden Wilder Mann, Abfahrt um 13:20 bis zur Haltestelle Hauptbahnhof
6. Überquerung der Straßenbahnschienen, 25 Meter geradeaus
7. Nach links wenden, 20 Meter geradeaus zum Eingang des Dresdner Hauptbahnhofes

Anschließend folgte der Außentest. Um die Anweisungen der Sprachausgabe verstehen zu können, kam zusätzlich ein Knochenleitkopfhörer der Firma Aftershokz² zum Einsatz. Bei diesem Kopfhörertyp wird der Schall über den Wangenknochen direkt in das Innenohr übertragen. Dadurch bleiben die Ohren des Trägers frei und behindern somit nicht die akustische Wahrnehmung der Umgebung.

Nun wurde die Route Punkt für Punkt abgelaufen. Bei den ersten zwei Kreuzungen erfolgte noch eine kurze Erläuterung der Daten der Routenverfolgungsoberfläche (siehe 4.5.4). Danach konnten alle Teilnehmer der Route bis zum Ziel folgen. Dies schließt auch die Fahrt mit den öffentlichen Verkehrsmitteln ein.

Teil 3: Auswertung und Nachbefragung

Im dritten Teil der Evaluation wurden die Probanden schließlich um eine Einschätzung des Prototypen gebeten.

Die Auswahl des Start- und Zielpunktes bewerteten alle Teilnehmer als leicht (auf einer Skala von 1 = sehr leicht bis 5 = sehr schwierig). Die in der zweiten Teilaufgabe gestellte Navigationsaufgabe bewältigten die Probanden alle mit leichten Unsicherheiten. Einem der Tester war die Route im Vorhinein bereits bekannt. Dem Zweiten waren Teilstücke der Route vertraut und die dritte Testperson hatte keinerlei Ortskenntnisse.

Die Ausführlichkeit der Routinginformationen schätzten zwei der Tester als gut, einer sogar als sehr gut ein. In diesem Zusammenhang wurden die dynamischen Informationen über Name und Richtung der Kreuzungsstraßen hervorgehoben (siehe 4.5.4 auf Seite 37).

²<http://www.aftershokz.com> abgerufen am 16.01.2014

Anhand dieser Informationen konnten sich die Probanden einen schnellen Überblick über den Aufbau der jeweiligen Kreuzung verschaffen.

Anschließend wurden die Teilnehmer gefragt, welche Attribute für sie eine barrierefreie bzw. angenehme Route ausmachen. Von Allen werden kleinere Straßen bevorzugt. Große Plätze und Kreuzungen sind hingegen zu vermeiden. Die akzeptierte Länge des in Kauf zu nehmenden Umwegs variiert allerdings von anderthalbfacher bis zu dreifacher Länge. Bei den ÖPNV Verbindungen werden diejenigen mit weniger Umstiegen favorisiert.

Derzeit werden die Abfahrts- und Ankunftszeiten der ÖPNV Verbindungen lediglich statisch in die Route eingebettet und nicht mehr aktualisiert. Die angegebenen Abfahrtszeiten für die Straßenbahn zum Hauptbahnhof wurde von keiner der Testpersonen erreicht. Dies ist zu einem großen Teil darauf zurückzuführen, dass die Probanden sich zunächst innerhalb des Prototypen zurecht finden mussten und teils noch Erklärungen zur Bedienung gegeben wurden, als die Route bereits berechnet und die Bahnverbindung somit schon ermittelt worden war. Dies konnte zum Teil dadurch kompensiert werden, dass die Nutzer den aktuellen Fahrplan der Straßenbahnhaltestelle abrufen konnten, sobald sie dort angekommen waren. Somit konnte auf eine spätere Bahnverbindung ausgewichen werden. Dennoch wünschte sich einer der Tester auch die automatische Aktualisierung der Verbindung innerhalb der Route.

Abschließend wurden die Probanden gebeten, den Navigationsprototypen mit ihren bisherigen Strategien zur Auffindung eines unbekanntes Ortes (siehe 6.2) zu vergleichen. Alle drei Tester bejahten den Mehrwert des Prototypen deutlich. Besonders hervorgehoben wurden in diesem Zusammenhang die Steigerung der Unabhängigkeit von Dritten sowie die direkte Integration des ÖPNV in die Route.

Alle Teilnehmer konnten sich vorstellen, das Programm künftig zum Erreichen unbekannter Orte einzusetzen.

6.3. ÖPNV Bewertungsfunktion

Im Rahmen des Nutzertests aus Kapitel 6.2 wurde bereits eine Route unter der Einbeziehung des ÖPNV errechnet und evaluiert. Von den Testern wurde sowohl die Art der Integration als auch die konkret gewählte Verbindung als positiv eingeschätzt. Allerdings handelte es sich dabei um eine einfache Direktverbindung zu der es nur wenige verkehrstechnische Alternativen gab. Daher eignet sich dieser Versuch nur bedingt zur Evaluation der Verbindungs-Bewertungsfunktion, die im Kapitel 4.5.3 definiert wurde.

In diesem Abschnitt wird deshalb eine zusätzliche Untersuchung der Bewertungsfunktion durchgeführt. Dabei soll ermittelt werden, ob die derzeitige Gewichtung der einzelnen Verbindungsteile den Nutzererwartungen entspricht. Zu diesem Zweck werden zunächst drei Testrouten anhand zufällig ausgewählter Start- und Zielpunkte erstellt. Im zweiten Schritt erfragt der prototypische Routingserver aus Kapitel 4.5.3 die nächsten Verbindungen und wählt anhand der aufgestellten Bewertungsfunktion die optimale Verbindung aus. Anschließend werden die abgerufenen Verbindungen der drei Routen einer Kontrollgruppe vorgelegt, die ihrerseits die optimale Verbindung auswählt. Schließlich werden die Ergebnisse verglichen.

6. Evaluation

Die drei Nutzertest-Teilnehmer erklärten sich bereit, auch an diesem Versuch mitzuwirken und die Kontrollgruppe zu stellen. Die Routen verlaufen innerhalb des Dresdner Stadtgebiets und die Verbindungsdaten stammen von der Deutschen Bahn. An dieser Stelle werden lediglich die Versuchsergebnisse präsentiert. Die einzelnen Verbindungen finden sich inklusive der Ausgaben der Bewertungsfunktion im Anhang A.4.

Die folgenden Routen wurden evaluiert:

- Route 1:
 - Start: Fakultät Informatik (Nöthnitzer Straße 46)
 - Ziel: Bahnhof Neustadt
 - Planungszeit: 16:50 Uhr

- Route 2:
 - Start: Bierstube (Bergstraße 51)
 - Ziel: Konzertlokation Beatpol (Altbriesnitz 2a)
 - Planungszeit: 16:02 Uhr

- Route 3:
 - Start: Cämmerswalder Straße 1
 - Ziel: An der Flutrinne 18
 - Planungszeit: 11:27 Uhr

Für die Routen 1 und 2 wurden jeweils 4 Verbindungen abgerufen, da sich für die darauf Folgenden keine neuen Varianten mehr ergaben. Sie unterschieden sich lediglich in den Abfahrtszeiten. Bei Route 3 wurde die fünfte Verbindung auf Grund der differierenden Abfahrtsstationen hingegen noch in die Bewertung aufgenommen.

In der Tabelle 6.1 werden die Ergebnisse präsentiert. Bei Route 1 entschieden sich alle Probanden für die zweite Verbindung. Bei Route 2 wurde ebenfalls einstimmig die dritte Verbindung ausgewählt. Bei der dritten Route waren zwei Tester für Verbindung 5 und einer für Verbindung 2. Die Bewertungsfunktion des Routingsservers ermittelte für Route 1 die Verbindung 2, für Route 2 die Verbindung 3 und für Route 3 die Verbindung 5.

Bei der Auswertung der Daten fällt auf, dass bei der Auswahl der optimalen Verbindung innerhalb der Kontrollgruppe fast immer Konsens herrscht. Lediglich bei Route 3 ist eine Abweichung zu beobachten. Dies deutet auf ähnliche Präferenzen bezüglich der optimalen Verbindung hin. Aus den Ergebnissen lässt sich ferner ableiten, dass sich die Probanden tendenziell für Verbindungen mit möglichst wenigen Umsteigevorgängen entscheiden. Bei den Schlussfolgerungen ist allerdings auch die geringe Teilnehmerzahl mit zu berücksichtigen.

Die Bewertungsfunktion für die ÖPNV-Verbindungen wählte bei allen drei Routen diejenige, die auch von der Kontrollgruppe favorisiert wird.

	Route 1	Route 2	Route 3
Verbindung 1	0	0	0
Verbindung 2	3 (*)	0	1
Verbindung 3	0	3 (*)	0
Verbindung 4	0	0	0
Verbindung 5	-	-	2 (*)

Tabelle 6.1.: Die Tabelle stellt die Ergebnisse der ÖPNV Verbindungsevaluation dar. Die Ziffern bezeichnen die Anzahl der Tester, welche sich für die jeweilige Verbindung entschieden haben. Die vom Routingserver errechnete optimale Verbindung wird durch den Stern gekennzeichnet.

6.4. Zusammenfassung

Dieser Abschnitt fasst die Ergebnisse des Evaluationskapitels zusammen und stellt sie den Anforderungen aus Kapitel 3.1 gegenüber.

A1.1: Während des Funktionstests wurden vier Routen ausgewählt und abgelaufen. Im Rahmen des Nutzertests kam eine weitere dazu, welche auch eine ÖPNV Verbindung enthielt und durch drei Testpersonen erfolgreich bewältigt wurde.

A1.2: Der gewünschte Wert für die Positionsgenauigkeit konnte nicht erreicht werden. Dennoch genügt das Maß an Genauigkeit, um die Routingaufgabe erfolgreich durchführen zu können, wie sich u.a. während des Nutzertests gezeigt hat. Präzisere GPS Daten würden die Lösung der Navigationsaufgabe allerdings weiter vereinfachen.

A1.3: Durch Verwendung von A-GPS ist die Planung der Route auch innerhalb von Gebäuden möglich (siehe Teilaufgabe 1 des Nutzertests). Es ist ferner möglich, sich mit der erstellten Route im Vorhinein vertraut zu machen.

A1.4: Routen können unter Einbeziehung des ÖPNV berechnet werden. Die Testpersonen schätzten diese Möglichkeit als sehr positiv ein. Die Bewertungsfunktion für die Wahl der optimalen Verbindung ist in Abschnitt 6.3 untersucht und für tauglich befunden worden.

A2.1: Das verwendete OSM Kartenmaterial erwies sich sowohl während der Funktions- wie auch beim Nutzertest als sehr ausführlich. Dies gilt insbesondere für kleine Fuß- und Feldwege. Weitere Kartenattribute werden berücksichtigt.

A2.2: Die Routen wurden um den Aufbau sowie Namen und Anzahl der Straßen einer Kreuzung ergänzt. Im Rahmen des Nutzertests wurde die Ausführlichkeit der Kreuzungsbeschreibung positiv hervorgehoben.

A2.3: Während der Implementierung des Routingserver wurde ein Skript zur Aktualisierung der lokalen OpenStreetMap Datenbank erstellt. Die Karten werden alle 14 Tage auf den aktuellen Stand gebracht.

A3: Die Zugänglichkeit des Clients wurde im Abschnitt 6.1 bestätigt.

7. Ausblick

Die positiven Ergebnisse von Nutzer- und Funktionstest geben Anlass zur Weiterentwicklung des prototypischen Systems.

Momentan ist die Berechnung von Routen auf das Gebiet von Sachsen beschränkt. Im nächsten Schritt wird statt dessen das OSM Kartenmaterial von Deutschland verwendet, um den Dienst deutschlandweit anbieten zu können.

Außerdem sollte untersucht werden, ob ein externer GPS Empfänger eine erheblich genauere Positionsbestimmung ermöglicht. Ein externer Empfänger sollte sich weiterhin positiv auf die Akkulaufzeit des Smartphones auswirken, da der interne Empfänger nicht eingesetzt werden muss.

Einige Attribute der OSM Karte werden bisher nur lückenhaft erfasst. Dazu gehören teils die Routen des ÖPNV, die konkreten Positionen der Ampelpfosten und besonders für Blinde interessante Eigenschaften wie beispielsweise Informationen über akustische Ampelanlagen. Daher wird eine engere Kooperation mit der OpenStreetMap Community angestrebt. In erster Linie wird versucht, den Fokus der Mapper verstärkt auf diese fehlenden Attribute zu lenken und die Motivation zur Erfassung der Eigenschaften und Relationen zu erhöhen. Des Weiteren ist geplant, den im Abschnitt 4.6 beschriebenen Ansatz zur Schaffung eines Querungsknotens den OSM Mitgliedern beispielsweise im Rahmen eines monatlich stattfindenden Stammtischs vorzustellen.

Um die Beschaffenheit eines komplexen Kreuzungsareals erfassen zu können, wird der Einsatz von taktilen Karten angeregt. Vorstellbar wäre, dass der Nutzer die Möglichkeit bekommt, im Vorhinein Teile der Route auszuwählen. Diese Teile könnten anschließend mit den Informationen über die Route aufbereitet und auf einem geeigneten Brailledrucker gedruckt werden.

Alternativ ist auch die Kopplung mit dem HyperBraille System vorstellbar¹. Im Rahmen des HyperBraille Projekts wird ein Braille-Flächendisplay zur Anzeige taktiler Grafiken entwickelt. Die Stiftplatten werden in zwei Größen gefertigt. Die Kleinere ist mit 32 x 30 Pins für den mobilen Einsatz konzipiert und eignet sich somit für die Ausgabe von Zusatzinformationen während des Routings. Des Weiteren wurde bereits eine Software entwickelt, welche einen Kartenausschnitt der OpenStreetMap bearbeitet und auf dem Braille-Display darstellt[18].

Vorstellbar wäre, das Projekt mit dem in dieser Arbeit vorgestellten Navigationsprototypen zu kombinieren. Beispielsweise könnte die Stiftplatte via Bluetooth mit dem Android Smartphone gekoppelt werden. Anschließend wäre es möglich, einen beliebigen Routenabschnitt auf dem taktilen Bildschirm darzustellen. Dazu müsste der HyperBraille Prototype allerdings um eine Bluetooth Schnittstelle erweitert und die verwendete

¹<http://www.hyperbraille.de> abgerufen am 21.01.2014

7. Ausblick

Software zur Darstellung der Karte auf das Android Betriebssystem portiert werden.

A. Anhang

A.1. Ausschnitt einer Fußgänger-Route

Im Folgenden wird das Teilstück einer vom Routingserver generierten Route dargestellt. Die Route liegt im Json Datenformat vor und stellt einen Ausschnitt aus der Testroute von Kapitel 6.2 dar. Die vollständige Route verläuft von der Fakultät Informatik der TU-Dresden zum Haupteingang des Dresdner Hauptbahnhofs. Der Ausschnitt zeigt die Teilstücke von der Kreuzung Nöthnitzer Straße, Georg-Schumann Straße bis zur Straßenbahnhaltestelle der Linie 3 am Hauptbahnhof. Der Übersichtlichkeit halber wurden einige Teile der Route entfernt, gekennzeichnet durch „...“.

```
1  [
2    {
3      ...
4    },
5    {
6      "sub_points": [
7        {
8          "intersection_bearing": 358,
9          "name": "Georg-Schumann-Straße",
10         "pois": [],
11         "lon": 13.7222136,
12         "surface": "Asphalt",
13         "node_id": 340071408,
14         "lat": 51.0263778,
15         "way_id": 4425970,
16         "sidewalk": 3,
17         "type": "footway",
18         "sub_type": "Wohnstraße"
19       },
20       {
21         "intersection_bearing": 285,
22         "name": "Nöthnitzer Straße",
23         "pois": [],
24         "lon": 13.7208801,
25         "surface": "Asphalt",
26         "node_id": 24939829,
27         "lat": 51.0263453,
```

A. Anhang

```
28         "way_id": 174342935,
29         "sidewalk": 1,
30         "type": "footway",
31         "sub_type": "Tertiärstraße"
32     },
33     {
34         "intersection_bearing": 102,
35         "name": "Nöthnitzer_ Straße",
36         "pois": [],
37         "lon": 13.7222751,
38         "surface": "Asphalt",
39         "node_id": 299837457,
40         "lat": 51.026104,
41         "way_id": 174342935,
42         "sidewalk": 1,
43         "type": "footway",
44         "sub_type": "Tertiärstraße"
45     }
46 ],
47 "name": "Nöthnitzer_ Straße, Georg-Schumann-Straße, ",
48 "traffic_signal_list": [],
49 "number_of_streets_with_name": 2,
50 "lon": 13.722224,
51 "turn": 73,
52 "node_id": 1274024020,
53 "lat": 51.0261113,
54 "type": "intersection",
55 "sub_type": "uncontrolled"
56 },
57 {
58     "bearing": 352,
59     "name": "Georg-Schumann-Straße",
60     "distance": 357,
61     "pois": [ ... ],
62     "surface": "Asphalt",
63     "way_id": 4425970,
64     "sidewalk": 3,
65     "type": "footway",
66     "sub_type": "Wohnstraße"
67 },
68 {
69     "sub_points": [ ... ],
70     "name": "Georg-Schumann-Straße, Münchner_ Platz, ",
71     "traffic_signal_list": [],
```

A. Anhang

```
72     "number_of_streets_with_name": 2,
73     "lon": 13.7215546,
74     "turn": 344,
75     "node_id": 15510135,
76     "lat": 51.0292969,
77     "type": "intersection",
78     "sub_type": "uncontrolled"
79   },
80   {
81     "bearing": 334,
82     "name": "Münchner_Platz",
83     "distance": 48,
84     "pois": [ ... ],
85     "surface": "Kopfsteinpflaster",
86     "way_id": 33616796,
87     "sidewalk": 3,
88     "type": "footway",
89     "sub_type": "Wohnstraße"
90   },
91   {
92     "sub_points": [ ... ],
93     "name": "Münchner_Straße, Münchner_Platz",
94     "traffic_signal_list": [
95       {
96         "distance": 10,
97         "name": "Ampel",
98         "bearing": 59,
99         "lon": 13.7213801,
100        "node_id": 268448952,
101        "lat": 51.0297357,
102        "type": "poi",
103        "sub_type": "Ampel"
104      }
105     ],
106     "number_of_streets_with_name": 2,
107     "lon": 13.721255,
108     "turn": 85,
109     "node_id": 5599375,
110     "lat": 51.0296898,
111     "type": "intersection",
112     "sub_type": "Ampel"
113   },
114   {
115     "bearing": 61,
```

A. Anhang

```
116     "name": "Münchner□Straße",
117     "distance": 63,
118     "pois": [],
119     "surface": "Asphalt",
120     "way_id": 4830078,
121     "sidewalk": 2,
122     "type": "footway",
123     "sub_type": "Tertiärstraße"
124 },
125 {
126     "name": "Münchner□Platz",
127     "lines": [
128         {
129             "to": "Dresden,□Wilder□Mann",
130             "nr": "3"
131         }
132     ],
133     "wheelchair": 1,
134     "lon": 13.7220637,
135     "station_id": 972115,
136     "node_id": 27306558,
137     "lat": 51.0300133,
138     "type": "station",
139     "sub_type": "Straßenbahnhaltestelle",
140     "accuracy": true
141 },
142 {
143     "direction": "Wilder□Mann,□Dresden",
144     "number_of_stops": 2,
145     "stops": [
146         "Nürnberger□Platz,□Dresden",
147         "Reichenbachstraße,□Dresden"
148     ],
149     "arrival_time": "18:53",
150     "duration": "5□Min",
151     "line": "T3",
152     "type": "transport",
153     "departure_time": "18:48"
154 },
155 {
156     "name": "Hauptbahnhof",
157     "lines": [
158         {
159             "to": "Dresden,□Wilder□Mann",
```

A. Anhang

```
160         "nr": "3"
161     },
162     {
163         "to": "Dresden , □ Hellerau " ,
164         "nr": "8"
165     }
166 ],
167     "wheelchair": 1,
168     "lon": 13.733539,
169     "station_id": 977992,
170     "node_id": 997588902,
171     "lat": 51.0398074,
172     "type": "station",
173     "sub_type": "Straßenbahnhaltestelle",
174     "accuracy": true
175 },
176 {
177     ...
178 }
179 ]
```

A.2. Abgelaufene Fußgängerrouen

Dieser Abschnitt beinhaltet die vier generierten Fußgängerrouen, welche im Rahmen des Funktionstests abgelaufen wurden (siehe Kapitel 6.1).

Route 1: Straßenbahnhaltestelle Pulsnitzer Straße -- Restaurant Saponi d'Italia

Die Route beginnt an der Straßenbahnhaltestelle Pulsnitzer Straße, verläuft etwa 20 Meter die Bautzner Straße entlang und biegt dann nach links in die Pulsnitzer Straße ein. Nach etwa 200 Metern biegt die Route nach links in die Louisenstraße ab, überquert die Talstraße und die Kamenzer Straße und endet am Restaurant Saponi d'Italia.

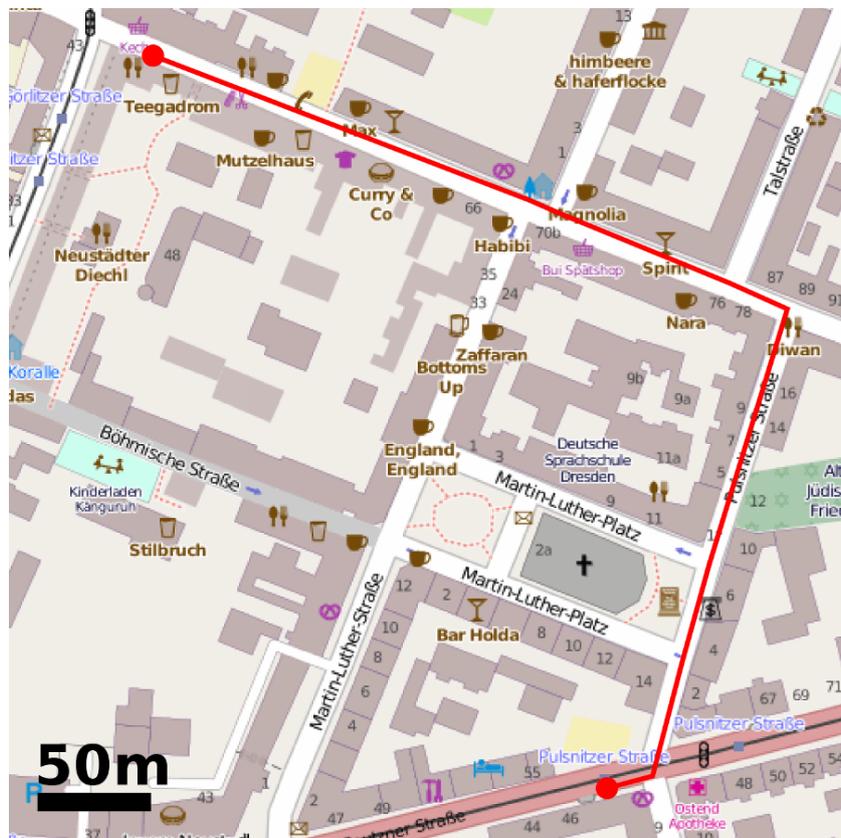


Abbildung A.1.: Die Route 1 verläuft von der Straßenbahnhaltestelle Pulsnitzer Straße zum Restaurant Saponi d'Italia. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

Route 2: Fakultät Informatik -- Cämmerswalder Straße 1

Die Route beginnt an der Fakultät Informatik, folgt der Nöthnitzer Straße etwa 100 Meter in östlicher Richtung und biegt dann nach rechts in einen Feldweg ab. Nach etwa 350 Metern folgt eine weitere Abzweigung nach rechts. Nach abermals reichlich 350 Metern biegt die Route leicht nach links ab und trifft schließlich auf die Cämmerswalder Straße.

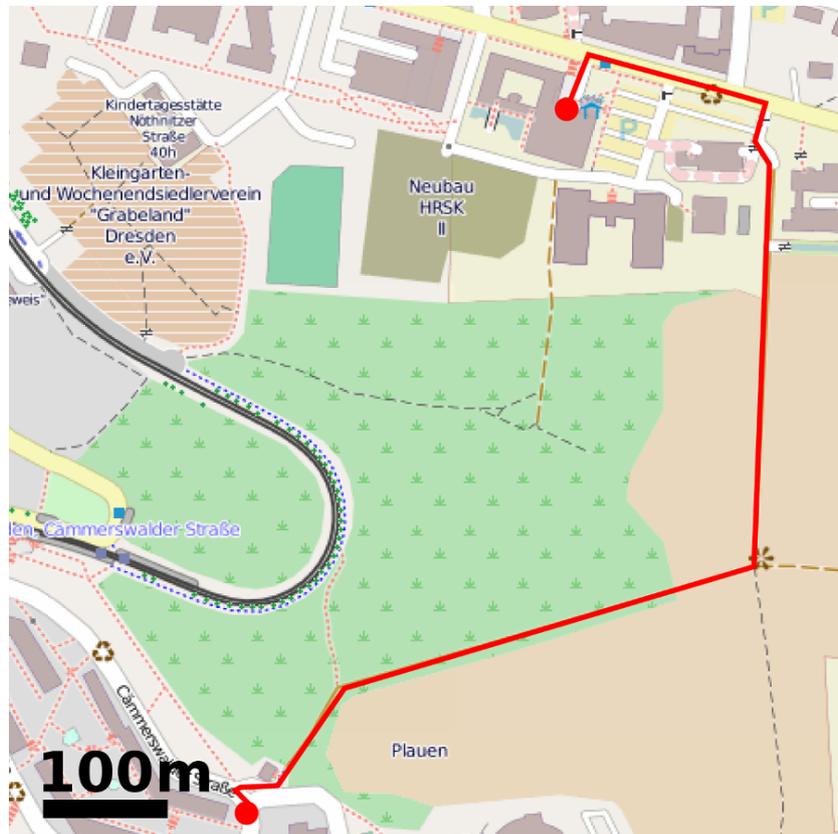


Abbildung A.2.: Die Route 2 verläuft von der Fakultät Informatik (Nöthnitzer Straße 46) zur Cämmerswalder Straße 1. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

Route 3: Cämmerswalder Straße 1 -- Rathaus Plauen

Start der Route ist die Cämmerswalder Straße 1. Die Route folgt der Cämmerswalder Straße, biegt nach etwa 100 Metern nach rechts und nach weiteren 40 Metern wieder nach links in einen Fußweg ab. Abermals nach etwa 100 Metern kreuzt die Route die Straßenbahnschienen und folgt dann circa 800 Meter dem Plauenschen Ring, bevor sie nach rechts in die hohe Straße abbiegt. Nach 150 Metern folgt die Abzweigung in die Nöthnitzer Straße, welche schließlich direkt zum Rathaus Plauen führt.

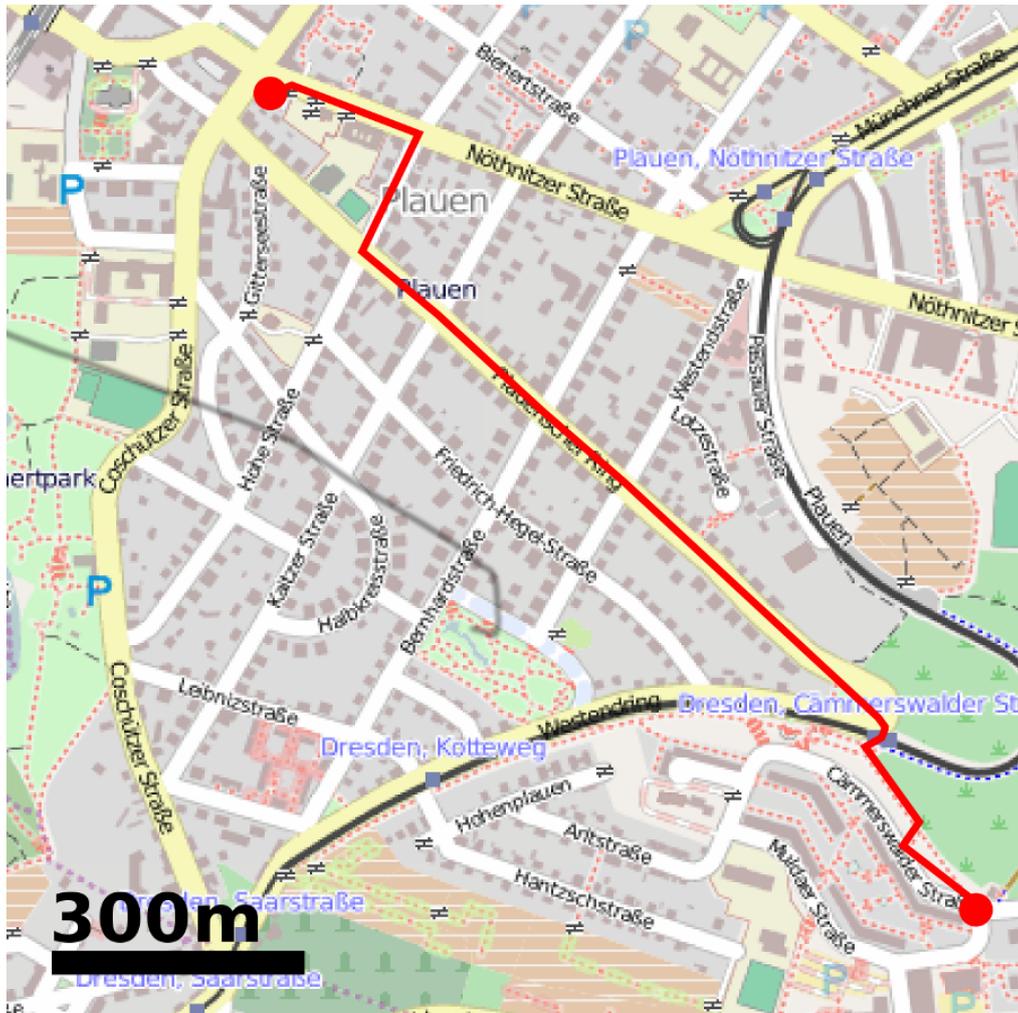


Abbildung A.3.: Die Route 3 verläuft von der Cämmerswalder Straße 1 zum Rathaus Plauen (Nöthnitzer Straße 2). Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

Route 4: Firat Kebap-Haus -- Eisenstückstraße 3

Das Firat Kebap-Haus bildet den Startpunkt. Die Route folgt der Mommsenstraße und biegt dann in die Helmholzstraße ab. Nach 400 Metern wird die Münchner Straße geradeaus überquert und nach weiteren 100 Metern folgt die Abzweigung in die Hübnerstraße. Nachdem auch die Nürnberger Straße geradeaus überquert wurde, biegt die Route schließlich nach links in die Eisenstückstraße ein und folgt ihr bis zur Hausnummer 3.

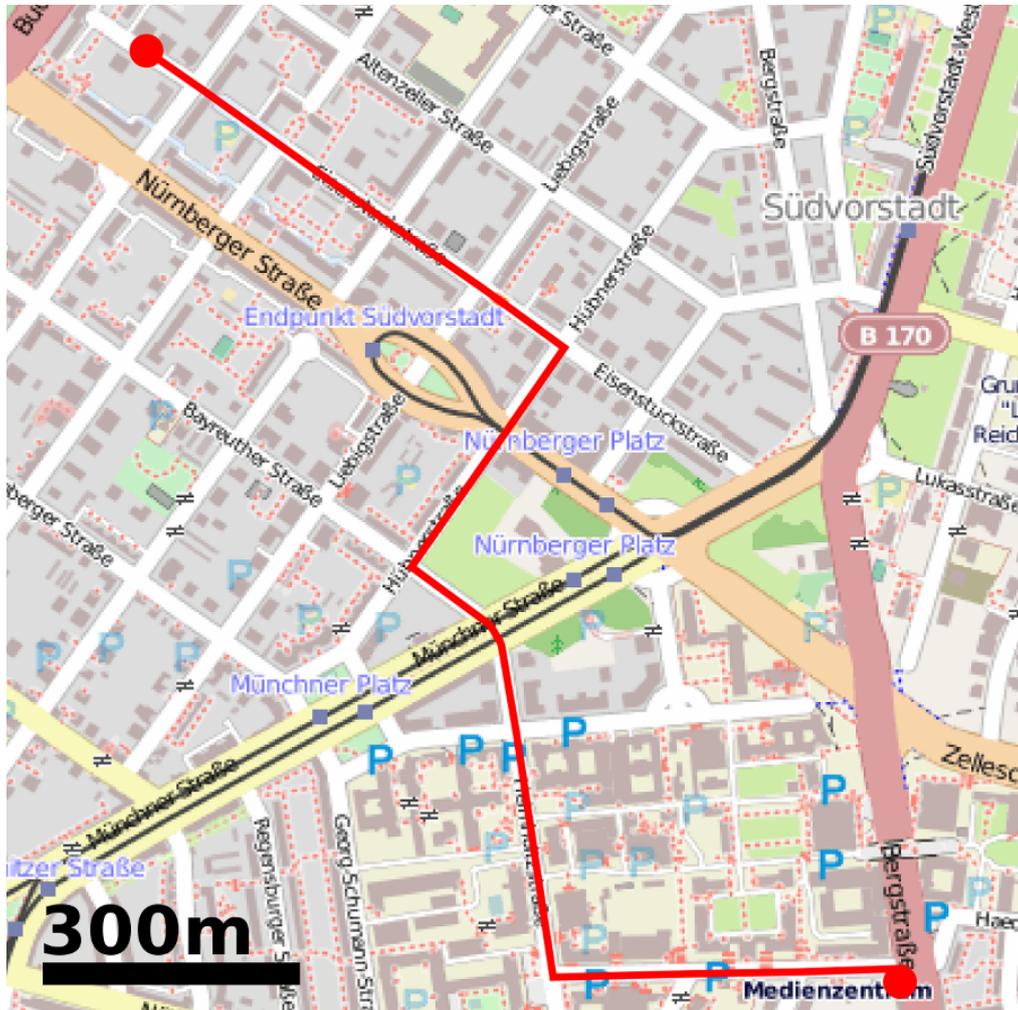


Abbildung A.4.: Die Route 4 verläuft vom Firat Kebap-Haus (Bergstraße 68) zur Eisenstückstraße 3. Quelle des Bildmaterials: ©OpenStreetMap und Mitwirkende, CC-BY-SA 2.0

A.3. Fragebogen für den Nutzertest

Der folgende Abschnitt enthält den Fragebogen, der im Rahmen des Nutzertests aus Kapitel 6.2 zur Befragung der Probanden verwendet wurde. Er gliedert sich in die Unterabschnitte Vor- und Nachbefragung.

Vorbefragung

1. Geburtsjahr:
2. Berufsgruppe/Ausbildung:
3. Wie ist der Grad Ihrer Sehbehinderung?
 - Sehbehindert
 - Hochgradig sehbehindert
 - Geburtsblind
 - Späterblindet im Alter von:
4. Verfügen Sie über ein Restsehvermögen? Wenn ja, inwiefern wirkt sich dies auf Ihre Orientierungsfähigkeiten aus?
5. Haben Sie weitere Einschränkungen der Wahrnehmungsfähigkeiten? (mehrere Antworten möglich)
 - Hörbehindert
 - Motorisch behindert
 - Einschränkung der Tastfähigkeit
 - Weitere Einschränkung
6. Wie hoch schätzen Sie Ihre Orientierungsfähigkeiten in unbekanntem Umgebungen ein?
 - In fremden Umgebungen kann ich mich sehr gut und ohne fremde Hilfe orientieren.
 - In fremden Umgebungen kann ich mich gut orientieren.
 - In fremden Umgebungen kann ich mich einigermaßen orientieren.
 - In fremden Umgebungen kann ich mich eher schlecht orientieren.
 - In fremden Umgebungen habe ich meist keine Orientierung, sodass ich auf fremde Hilfe angewiesen bin.
7. Benutzen Sie mobile Geräte wie Handy und Smartphone?
 - Wie oft?

A. Anhang

- Seit wann?
- Einschätzung des Erfahrungsgrades:
 - Gering
 - Mittel
 - Hoch

8. Angenommen, Sie müssen einen unbekanntem Ort erreichen. Wie gehen Sie dabei vor und welche Hilfsmittel kommen zum Einsatz?

9. Haben Sie bereits Erfahrungen mit Navigations-Programmen gesammelt?

- Falls ja, welche:
- Falls nein, warum nicht?
 - Kein Handy oder Smartphone vorhanden
 - Noch nicht ausprobiert
 - Für untauglich befunden

Nachbefragung

1. Wie schwer fiel Ihnen die Auswahl des Start- und Zielpunktes (Teilaufgabe 1)?

- Sehr leicht
- Leicht
- Mittelmäßig
- Schwer
- Sehr schwer

2. Konnten Sie die Navigationsaufgabe bewältigen (Teilaufgabe 2)?

- Ohne Probleme
- Mit leichten Unsicherheiten
- Mit Mühe
- Nein

3. War Ihnen die Testroute bereits im Vorhinein vertraut?

- Ja
- Teilweise

A. Anhang

- Nein
4. Bitte bewerten Sie die Ausführlichkeit der Routinganweisungen.
- Sehr ausführlich: Alle benötigten Informationen über die Beschaffenheit der nächsten Routenpunkte waren vorhanden.
 - Ausführlich: Ich konnte der Route ohne größere Einschränkungen folgen. Nur an einigen wenigen Stellen wären zusätzliche Angaben hilfreich gewesen.
 - Mittelmäßig: Die wichtigen Informationen über den Routenverlauf waren vorhanden. Allerdings fehlten einige für mich relevante Anweisungen.
 - Wenig ausführlich: Der Route konnte ich nur mit Mühe folgen, da wesentliche Teile der Routinganweisungen fehlten.
 - Ungenügend: Das Verfolgen der Route war mir anhand der vorhandenen Informationen nicht möglich.
5. Bitte vergleichen Sie den getesteten Prototypen mit Ihren bisher verwendeten Navigationstechniken.
6. Welche Kriterien machen für Sie eine angenehme Route aus und was versuchen Sie zu vermeiden?

A.4. Ergebnis der Evaluation der ÖPNV-Routen

Dieser Abschnitt stellt die Ergebnisse der Evaluation der ÖPNV-Routen dar. Nach dem Abrufen der Routen wird jede Verbindung anhand der Regeln aus Kapitel 4.5.3 bewertet.

Route 1: Fakultät Informatik - Bahnhof Neustadt

Planung: 16:50

Verbindung 1:

- Fußweg = 306 Meter: 3 Punkte
- 23 Minuten bis Abfahrt: 5 Punkte
- 17:13 - 17:15 -- line: B85; From Helmholtzstraße to Plauen, Nöthnitzer Straße
- Umsteigen: Genaue Haltestellenposition: Ja, Unterschiedliche Haltestelle, Umsteigezeit zu knapp: 106 Punkte
- 17:17 - 17:35 -- line: T3; From Plauen, Nöthnitzer Straße to Bahnhof Neustadt
- Fahrtdauer = 22 Minuten: 3 Punkte
- Gesamt: 117 Punkte

A. Anhang

Verbindung 2:

- Fußweg = 595 Meter, 5 Punkte
- 28 Minuten bis Abfahrt: 6 Punkte
- 17:18 - 17:35 -- line: T3; From Münchner Platz to Bahnhof Neustadt
- Fahrdauer = 17 Minuten: 2 Punkte
- Gesamt: 13 Punkte

Verbindung 3:

- Fußweg = 895 Meter, 8 Punkte
- 36 Minuten bis Abfahrt: 8 Punkte
- 17:26 - 17:29 -- line: T8; From Nürnberger Platz to Hauptbahnhof
- Umsteigen: Wechsel der Verkehrsmittelklasse, Umsteigezeit zu knapp: 108 Punkte
- 17:37 - 17:42 -- line: RB17285; From Hauptbahnhof (Gleis 11) To Bahnhof Neustadt (Gleis 5)
- Fahrdauer = 16 Minuten: 2 Punkte
- Gesamt: 126 Punkte

Verbindung 4:

- Fußweg = 595 Meter, 5 Punkte
- 38 Minuten bis Abfahrt: 8 Punkte
- 17:28 - 17:45 -- line: T3; From Münchner Platz to Bahnhof Neustadt
- Fahrdauer = 17 Minuten: 2 Punkte
- Gesamt: 15 Punkte

Route 2: Bierstube - Beatpol Dresden

Planung: 16:02

Verbindung 1:

- Fußweg = 313 Meter: 3 Punkte
- 11 Minuten bis Abfahrt: 3 Punkte
- 16:13 - 16:17 -- line: B360; From Technische Universität, Dresden to Dresden Hauptbahnhof

A. Anhang

- Umsteigen: Genaue Haltestellenposition: Nein, unterschiedliche Haltestellen-Id, Umsteigezeit zu knapp: 106 Punkte
- 16:21 - 16:26 -- line: T10; From Hauptbahnhof to Bahnhof Mitte / Jahnstraße
- Umsteigen: Genaue Haltestellenposition: Nein, gleiche Haltestellen-Id, Umsteigezeit zu knapp: 104 Punkte
- 16:29 - 16:42 -- line: B94; From Bahnhof Mitte, Dresden to Schunckstraße, Dresden
- Fahrdauer = 29 Minuten: 3 Punkte
- Gesamt: 219 Punkte

Verbindung 2:

- Fußweg = 634 Meter: 6 Punkte
- 16 Minuten bis Abfahrt: 4 Punkte
- 16:18 - 16:26 -- line: B61; From Technische Universität (Fritz-Foerster-Platz) to Tharandter Straße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 16:31 - 16:43 -- line: T12; From Tharandter Straße to Cossebauder Straße
- Fahrdauer = 25 Minuten: 3 Punkte
- Gesamt: 15 Punkte

Verbindung 3:

- Fußweg = 574 Meter: 5 Punkte
- 16 Minuten bis Abfahrt: 4 Punkte
- 16:18 - 16:26 -- line: B61; From Technische Universität (Fritz-Foerster-Platz) to Tharandter Straße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 16:31 - 16:44 -- line: T12; From Tharandter Straße to Gottfried-Keller-Straße
- Fahrdauer = 26 Minuten: 3 Punkte
- Gesamt: 14 Punkte

Verbindung 4:

- Fußweg = 634 Meter: 6 Punkte

A. Anhang

- 28 Minuten bis Abfahrt: 6 Punkte
- 16:30 - 16:38 -- line: B61; From Technische Universität (Fritz-Foerster-Platz) to Tharandter Straße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 16:41 - 16:53 -- line: T12; From Tharandter Straße to Cossebauder Straße
- Fahrdauer = 23 Minuten: 3 Punkte
- Gesamt: 17 Punkte

Route 3: Cämmerswalder Straße 1 - An der Flutrinne 18

Planung: 11:27

Verbindung 1:

- Fußweg = 449 Meter: 4 Punkte
- 9 Minuten bis Abfahrt: 2 Punkte
- 11:36 - 11:39 -- line: B63; From Dorfhainer Straße, Dresden to Dresden, Saarstraße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 11:43 - 11:55 -- line: T3; From Dresden, Saarstraße to Walpurgisstraße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 11:57 - 12:16 - line: T9; From Walpurgisstraße to An der Flutrinne, Dresden
- Fahrdauer = 40 Minuten: 5 Punkte
- Gesamt: 15 Punkte

Verbindung 2:

- Fußweg = 449 Meter: 4 Punkte
- 21 Minuten bis Abfahrt: 5 Punkte
- 11:48 - 11:59 -- line: B66; From Dorfhainer Straße, Dresden to Gret-Palucca-Straße, Dresden
- Umsteigen: Genaue Haltestellenposition: Nein, gleiche Haltestellen-Id: 4 Punkte
- 12:04 - 12:26 -- line: T9; From Gret-Palucca-Straße to An der Flutrinne
- Fahrdauer = 38 Minuten: 4 Punkte

A. Anhang

- Gesamt: 17 Punkte

Verbindung 3:

- Fußweg = 449 Meter: 4 Punkte
- 29 Minuten bis Abfahrt: 6 Punkte
- 11:56 - 11:59 -- line: B63; From Dorfhainer Straße, Dresden to Dresden, Saarstraße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 12:03 - 12:15 -- line: T3; From Dresden, Saarstraße to Walpurgisstraße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 12:17 - 12:36 - line: T9; From Walpurgisstraße to An der Flutrinne, Dresden
- Fahrtdauer = 40 Minuten: 5 Punkte
- Gesamt: 19 Punkte

Verbindung 4:

- Fußweg = 449 Meter: 4 Punkte
- 41 Minuten bis Abfahrt: 9 Punkte
- 12:08 - 12:19 -- line: B66; From Dorfhainer Straße, Dresden to Gret-Palucca-Straße, Dresden
- Umsteigen: Genaue Haltestellenposition: Nein, gleiche Haltestellen-Id: 4 Punkte
- 12:24 - 12:46 -- line: T9; From Gret-Palucca-Straße to An der Flutrinne
- Fahrtdauer = 38 Minuten: 4 Punkte
- Gesamt: 21 Punkte

Verbindung 5:

- Fußweg = 367 Meter: 3 Punkte
- 18 Minuten bis Abfahrt: 4 Punkte
- 11:45 - 11:55 -- line: T3; From Dresden, Cämmerswalder Straße to Walpurgisstraße
- Umsteigen: Genaue Haltestellenposition: Ja, gleiche Haltestelle: 2 Punkte
- 11:57 - 12:16 - line: T9; From Walpurgisstraße to An der Flutrinne, Dresden
- Fahrtdauer = 31 Minuten: 4 Punkte
- Gesamt: 13 Punkte

Literaturverzeichnis

- [1] I Ausserer, K Ausserer, O Krammer, D Reiter, A Risser, R Risser, and D Wunsch. *Telematik im Öffentlichen Verkehr: Eine Möglichkeit zu mehr Chancengleichheit im Verkehr für Menschen mit Körper-und Sinnesbeeinträchtigungen*, chapter 4.2.6. 2006.
- [2] Sarah Chiti and Barbara Leporini. Accessibility of android-based mobile devices: a prototype to investigate interaction with blind users. In *Computers Helping People with Special Needs*, pages 607--614. Springer, 2012.
- [3] Graeme Douglas and Brazil). International Council for Education of the Visually Handicapped. Conference (10th: 1997: Sao Paulo. *MoBIC: a system for facilitating independent mobility and navigation for blind people: paper presented at ICEVH 10th World Conference, Sao Paulo, Brazil, 3-8 August, 1997*. University of Birmingham, Research Centre for the Education of the Visually Handicapped, 1997.
- [4] Marco Helbich, Christof Amelunxen, Pascal Neis, and Alexander Zipf. Investigations on locational accuracy of volunteered geographic information using openstreetmap data. In *GIScience 2010 Workshop, Zurich, Switzerland*, pages 14--17, 2010.
- [5] Daniel Hänßgen. Ein orientierungssystem für menschen mit sehbehinderung auf java me: Konzeption und implementierung. Master's thesis, Fachhochschule Hannover, 2010.
- [6] Jimmy LaMance, Javier DeSalas, and Jani Jarvinen. Assisted gps: a low-infrastructure approach. *GPS World*, 13(3), 2002.
- [7] Oihana Otaegui, Estíbaliz Loyo, Eduardo Carrasco, Mag. Caludia Fösleitner, John Spiller, Daniela Patti, Rafael Olmedo, and Markus Dubielzig. Argus: Assisting personal guidance system for people with visual impairment. 2012.
- [8] Thomas STROTHOTTE¹ Helen PETRIE, Valerie JOHNSON, and Lars REICHERT. Mobic: user needs and preliminary design for a mobility aid for blind and elderly travellers. In *The European Context for Assistive Technology: Proceedings of the 2nd Tide Congress, 26-28 April 1995, Paris*, volume 1, page 348. IOS Press, 1995.
- [9] Stuart Russell and Peter Norvig. *Künstliche Intelligenz -- Ein moderner Ansatz*, volume 2. Pearson Studium, 2004.

- [10] Thomas Strothotte, Steffi Fritz, Rainer Michel, Andreas Raab, Helen Petrie, Valerie Johnson, Lars Reichert, and Axel Schalt. Development of dialogue systems for a mobility aid for blind people: initial design and usability testing. In *Proceedings of the second annual ACM conference on Assistive technologies*, pages 139–144. ACM, 1996.
- [11] Henny Swan. Making android apps voice output accessible. 2012. Online; Accessed 2014-01-20.
- [12] InMoBS Team. Auswertung der inmobs-fragebogenaktion. 2013. Online; Accessed 2013-12-30.
- [13] DBSV u. a. ; Fachausschuss für Informations- und Telekommunikationssysteme. Anforderungen an satellitengestützte navigationssysteme für blinde und sehbehinderte menschen. 2008. Online; accessed 2013-08-20.
- [14] Thorsten Völkel. *Multimodale Annotation geographischer Daten zur personalisierten Fußgängernavigation*. PhD thesis, Dresden University of Technology, 2008.
- [15] Peter Wendor. Explorative evaluierung von navigationsassistentz für menschen mit behinderungen auf basis von openstreetmap. Master’s thesis, Universität Paderborn, Fakultät für Elektrotechnik, Informatik und Mathematik, 2010.
- [16] OpenStreetMap Wiki. Routing/online routers --- openstreetmap wiki,, 2013. [Online; accessed 2013-08-20.
- [17] J WOELFL, E LEUPRECHT, U SIGL, M HRUNEK, M BAUER, and K HASELBOECK. Unterwegs im dunkeln. forschungsbericht ueber die mobilitaetsbedingungen von blinden und sehbehinderten personen in wien unter besonderer beruecksichtigung des oeffentlichen personennahverkehrs. 2004.
- [18] Limin Zeng, Gerhard Weber, and Ulrich Baumann. Audio-haptic you-are-here maps on a mobile touch-enabled pin-matrix display. In *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*, pages 95–100. IEEE, 2012.
- [19] Dennis Zielstra and Alexander Zipf. A comparative study of proprietary geodata and volunteered geographic information for germany. In *13th AGILE International Conference on Geographic Information Science*, volume 2010, 2010.

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig, unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Dresden, den 04.02.2014