Karlsruher Institut für Technologie

An Orientation & Mobility Aid for People with Visual Impairments

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation
von

Daniel Koester

aus Stralsund, Deutschland

Tag der mündlichen Prüfung:	22. Mai 2019
Hauptreferent:	Prof. Dr.-Ing. Rainer Stiefelhagen Karlsruher Institut für Technologie
Korreferent:	a.Univ.-Prof. Dr. Klaus Miesenberger Johannes Kepler Universität Linz

Abstract

Orientation & Mobility (O&M) encompasses a range of techniques for people with visual impairments that teach them how to roam around in everyday situations. However, they have to undergo extensive and laborious personal training sessions with O&M specialists, to ingrain these techniques into their daily routines. While some techniques rely on assistive devices, such as the popular White Cane, Points of Interest databases or compass-based orientation systems, there is an inconspicuous communication disconnect between available guidance and navigation techniques.

Within recent years, truly mobile computing platforms — smartphones — have become ubiquitous. This potentially allows modern computer vision techniques to assist the human visual system on issues that arise when everyday objects and situations are not designed with proper accessibility in mind. However, special care has to be taken in order to not conflict with specific individually learned behaviors or even contradict O&M techniques in these situations.

In this thesis we identify a spatial and systemic gap between guidance and navigation aids for people with visual impairments. This spatial gap exists largely because assistive guidance devices, *i.e.*, the White Cane, can only perceive the surroundings within a certain range, while navigation information only provides instructions on a very broad scale. Additionally, it is partly caused by the systemic gap between these two components — the White Cane never considers routing, while a navigation system does not know about nearby obstacles or O&M techniques. Therefore, we try to diminish this disconnect and propose several approaches to increase interconnectivity between guidance and navigation information, approaching the issue from both directions. To ultimately provide always relevant and beneficial information only, we first identify the major requirements of assistive systems and create some key concepts that guide all of our algorithmic approaches and prototype implementations.

Current orientation assistance is mostly based on Global Navigation Satellite System applications. We try to improve these by proposing an approach that is capable of performing an individualized, impairment aware, and shoreline level-based routing. Generated routes are, while exhibiting a slightly increased travel distance, much safer according to our pre-defined objective factors created in accordance with O&M trainers. Furthermore, we improve the availability of relevant geospatial data required for such an impairment aware routing. To this end, we propose a state of the art data driven algorithm to detect zebra crossings in aerial imagery, which also works reliably across country borders.

To increase the usefulness of mobility assistance that is provided by computer vision methods, we propose approaches that are closely modeled after O&M techniques, in order to increase spatial awareness of the immediate surroundings. First, we detect the accessible section in front of the user and also inform about possible obstacles. Then, we create a novel approach to detect and precisely locate shorelines in the immediate vicinity and are able to create virtual segments that bridge shoreline discontinuities, providing relevant information about the next segment's location in advance. Finally, we improve the accessibility of pedestrian crossing situations, *i.e.*, zebra crossings and pedestrian traffic lights, using a deep learning-based approach.

To analyze whether our proposed approaches and algorithms truly benefit people with visual impairments, we perform a small — nonetheless very encouraging — Wizard of Oz experiment about our shoreline level routing. Furthermore, we conduct a much larger user study that combines different components, focused on pedestrian crossing scenarios. While our statistical measures show only little improvements so far, mostly due to initial prototype issues and little training time for participants to get sufficiently accustomed to it, we receive very encouraging comments by almost all study participants. Although much further work is required, this suggests that we have created an important first step towards diminishing the identified gap and already improved Orientation & Mobility for people with visual impairments.

Kurzzusammenfassung

Orientierung&Mobilität (O&M) umfasst eine Reihe von Techniken für Menschen mit Sehschädigungen, die ihnen helfen, sich im Alltag zurechtzufinden. Dennoch benötigen sie einen umfangreichen und sehr aufwendigen Einzelunterricht mit O&M Lehrern, um diese Techniken in ihre täglichen Abläufe zu integrieren. Während einige dieser Techniken assistive Technologien benutzen, wie zum Beispiel den Blinden-Langstock, Points of Interest Datenbanken oder ein Kompass gestütztes Orientierungssystem, existiert eine unscheinbare Kommunikationslücke zwischen verfügbaren Hilfsmitteln und Navigationssystemen.

In den letzten Jahren sind mobile Rechensysteme, insbesondere Smartphones, allgegenwärtig geworden. Dies eröffnet modernen Techniken des maschinellen Sehens die Möglichkeit, den menschlichen Sehsinn bei Problemen im Alltag zu unterstützen, die durch ein nicht barrierefreies Design entstanden sind. Dennoch muss mit besonderer Sorgfalt vorgegangen werden, um dabei nicht mit den speziellen persönlichen Kompetenzen und antrainierten Verhaltensweisen zu kollidieren, oder schlimmstenfalls O&M Techniken sogar zu widersprechen.

In dieser Dissertation identifizieren wir eine räumliche und systembedingte Lücke zwischen Orientierungshilfen und Navigationssystemen für Menschen mit Sehschädigung. Die räumliche Lücke existiert hauptsächlich, da assistive Orientierungshilfen, wie zum Beispiel der Blinden-Langstock, nur dabei helfen können, die Umgebung in einem limitierten Bereich wahrzunehmen, während Navigationsinformationen nur sehr weitläufig gehalten sind. Zusätzlich entsteht diese Lücke auch systembedingt zwischen diesen beiden Komponenten — der Blinden-Langstock kennt die Route nicht, während ein Navigationssystem nahegelegene Hindernisse oder O&M Techniken nicht weiter betrachtet. Daher schlagen wir verschiedene Ansätze zum Schließen dieser Lücke vor, um die Verbindung und Kommunikation zwischen Orientierungshilfen und Navigationsinformationen zu verbessern und betrachten das Problem dabei aus beiden Richtungen. Um nützliche relevante Informationen bereitzustellen, identifizieren wir zuerst die bedeutendsten Anforderungen an assistive Systeme und erstellen einige Schlüsselkonzepte, die wir bei unseren Algorithmen und Prototypen beachten.

Existierende assistive Systeme zur Orientierung basieren hauptsächlich auf globalen Navigationssatellitensystemen. Wir versuchen, diese zu verbessern, indem wir einen auf Leitlinien basierenden Routing Algorithmus erstellen, der auf individuelle Bedürfnisse anpassbar ist und diese berücksichtigt. Generierte Routen sind zwar unmerklich länger, aber auch viel sicherer, gemäß den in Zusammenarbeit mit O&M Lehrern erstellten objektiven Kriterien. Außerdem verbessern wir die Verfügbarkeit von relevanten georeferenzierten Datenbanken, die für ein derartiges bedarfsgerechtes Routing benötigt werden. Zu diesem Zweck erstellen wir einen maschinellen Lernansatz, mit dem wir Zebrastreifen in Luftbildern erkennen, was auch über Ländergrenzen hinweg funktioniert, und verbessern dabei den Stand der Technik.

Um den Nutzen von Mobilitätsassistenz durch maschinelles Sehen zu optimieren, erstellen wir O&M Techniken nachempfundene Ansätze, um die räumliche Wahrnehmung der unmittelbaren Umgebung zu erhöhen. Zuerst betrachten wir dazu die verfügbare Freifläche und informieren auch über mögliche Hindernisse. Weiterhin erstellen wir einen neuartigen Ansatz, um die verfügbaren Leitlinien zu erkennen und genau zu lokalisieren, und erzeugen virtuelle Leitlinien, welche Unterbrechungen überbrücken und bereits frühzeitig Informationen über die nächste Leitlinie bereitstellen. Abschließend verbessern wir die Zugänglichkeit von Fußgängerübergängen, insbesondere Zebrastreifen und Fußgängerampeln, mit einem Deep Learning Ansatz.

Um zu analysieren, ob unsere erstellten Ansätze und Algorithmen einen tatsächlichen Mehrwert für Menschen mit Sehschädigung erzeugen, vollziehen wir ein kleines Wizard-of-Oz-Experiment zu unserem bedarfsgerechten Routing — mit einem sehr ermutigendem Ergebnis. Weiterhin führen wir eine umfangreichere Studie mit verschiedenen Komponenten und dem Fokus auf Fußgängerübergänge durch. Obwohl unsere statistischen Auswertungen nur eine geringfügige Verbesserung aufzeigen, beeinflußt durch technische Probleme mit dem ersten Prototypen und einer zu geringen Eingewöhnungszeit der Probanden an das System, bekommen wir viel versprechende Kommentare von fast allen Studienteilnehmern. Dies zeigt, daß wir bereits einen wichtigen ersten Schritt zum Schließen der identifizierten Lücke geleistet haben und Orientierung&Mobilität für Menschen mit Sehschädigung damit verbessern konnten.

Acknowledgements

First and foremost, I would like to thank Prof. Rainer Stiefelhagen for all his guidance and supervision, giving me the opportunity to acquire a deep understanding of the needs of people with visual impairments, and the chance to try and make their lives a little bit easier and safer. I also want to thank Klaus Miesenberger for agreeing to be a reviewer of this thesis and for contributing much valued feedback to it.

Many thanks go to all former and current colleagues for their collaboration, their support, endless hours of insightful discussions, and all the fun times at the lab: Alina, Angela, Anke, Boris, Gerhard, Karin, Lukas, Makarand, Manel, Martin, Monica, Saquib, Thorsten, Tobias G., Tobias R., Vanessa, Vivek, and Ziad. Special thanks to Corinna for her unrelenting aid and navigation of any bureaucratic obstacles of the university's administration apparatus, you truly were my savior. Furthermore, thanks to Heiko, Lothar, Matthias, and Olaf — easing the burden of technical administration.

Moreover, I would like to thank all supervised students, including: Benjamin, Björn, Daniel, Dario, Maximilian, Mike, Paul, Tabea and Tobias as well as the numerous participants of my practical courses. I hope it's been an interesting learning experience for all of you as much as it has been for myself — it has provided me with an innumerable amount of human and scientific insights and ideas.

Most importantly, I would like to thank my parents and family for always believing in me, as well as my long-time girlfriend — *soon-to-be wife* — Nicole for having the patience and understanding for this daring endeavor of mine and for all her support over the years, as well as my son Tim and — *just very recently born* — daughter Lillie for reminding me of the truly important things in life and showing me the awesomeness of the world as seen through a kid's eyes.

Thank you all and also those that I might have forgotten to list here!

Contents

List of Abbreviations & Glossary

APS Accessible Pedestrian Signal . . . 60
Pedestrian walk light signals that exhibit additional accessibility features, *i.e.*, haptic or acoustic mechanisms, pilot tones, and ideally, relief symbols.

BLE Bluetooth Low Energy . . . 16
A wireless personal area network protocol with reduced power consumption at a similar communication range when compared to standard Bluetooth.

DGPS Differential Global Positioning System . . . 14
Provides improved positioning accuracy for GPS systems where a fixed receiver with a precisely known location sends out its positioning error to nearby devices.

GIS Geographic Information System . . . 14
Tools and databases to create, edit, and further process location related data for objects of all kinds, *e.g.*, longitude and latitude of pedestrian crossings.

GNSS Global Navigation Satellite System . . . 7
Provides a signal receiver's location — longitude, latitude, and altitude — based on timed signals (line of sight) to intermediate circular orbit satellites.

GPS Global Positioning System . . . 19
The first fully operational GNSS, created by the United States Department of Defense, providing a location accuracy of ∼10m to the general public since 2000.

GPU Graphical Processing Unit . . . 6
Specialized hardware originally intended for computer graphics and user interfaces, in recent years a driving factor for deep learning research and its applications.

HOG Histograms of Oriented Gradients . . . 45
A feature descriptor that represents local appearance in connected image regions using a histogram over intensity gradients, mostly used for object detection.

IMU Inertial Measurement Unit . . . 12
An electronic component measuring its own movements, *i.e.*, linear accelerations, using per axis accelerometers and gyroscopes, today a part of every smartphone.

LBP Local Binary Pattern . . . 45
A feature descriptor that creates a histogram over neighboring pixels compared to their center point value, encoded as zeros and ones, usually used for classification.

O&M Orientation & Mobility . . . 1
A set of techniques (as well as a profession, also known as O&M trainers) to assist people with visual impairments in autonomous and self-sufficient travel.

OSM OpenStreetMap . . . 42
A collaborative project by a community of volunteers that create a map of the world and provide open access to all its data. https://www.openstreetmap.org

OSRM Open Source Routing Machine . . . 57
A customizable, free, and open source routing engine usable as a web service or library that relies on OSM road network data. http://project-osrm.org

POI Points of Interest . . . 16
A coordinate pair of a specific location or object of interest, sometimes as part of a GIS system, often used in GNSS contexts.

RANSAC Random Sample Consensus . . . 25
An iterative model fitting algorithm that produces a non-deterministic result by randomly sampling a set of points (often with noise), reducing outlier influence.

RBF Radial Basis Function . . . 46
A kernel function that models the distance of two points within its decision region, often used for classification with SVMs.

RFID Radio-Frequency Identification . . . 15
A wireless and low-power, passive or active, identification method of physical objects based on electromagnetic fields that does not require a line of sight.

ROC Receiver Operating Characteristic . . . 49
A graphical curve that represents the performance of a binary classifier when modifying its decision threshold, comparing true and false positive rates.

ROI Region of Interest . . . 26
A specific range of data within a much larger set, often used to reduce processing time, generally similar to a POI but for a whole region instead.

SVM Support Vector Machine . . . 24
A machine learning model that classifies data into two disjunct categories, usually mapping features to a higher dimensional space that allows for better separation.

TERRAIN Selbständige Mobilität blinder und sehbehinderter Menschen im urbanen Raum durch audio-taktile Navigation . . . 110
A three-year BMBF project to improve O&M of people with visual impairments in urban areas with a mobile assistive system. https://www.terrain-projekt.de

TOF Time-of-Flight . . . 12
A measurement of the time it takes an object, *i.e.*, a particle or wave, to travel a certain distance within a medium, *e.g.*, used in cameras to collect depth data.

VGI Volunteered Geographical Information . . . 19
Voluntarily collected information about one's surroundings and other places, often assembled into large GIS databases for a multitude of purposes.

XML Extensible Markup Language . . . 43
A markup language that tries to be human-readable as well as machine-readable, by encoding information, especially data structures, into a structured format.

YOLO You Only Look Once . . . 98
A deep learning framework for object detection and classification that we use as part of our TERRAIN prototype.

List of Figures

List of Tables

Chapter 1

Introduction

"The only thing worse than being blind is having sight but no vision."

– Helen Keller

The concept of *Orientation&Mobility (O&M)* was developed by Richard Hoover and Russel Williams in 1944, latter being blinded by enemy action in France, who were then later joined by Warren Bledsoe in 1947, at the Valley Forge Army General Hospital. Their original intention was to aid veterans that suffered visual impairments from World War II in their daily guidance and navigation tasks (Wiener et al., 2010). Through blindfolded trial and error, Hoover came up first with the original idea to move a very lightweight cane in an arc in front of the body, always touching the ground on the side opposite of the forward foot, a style that is still dominant and practiced by professional O&M trainers.

Orientation&Mobility today has become a set of *de facto* training techniques that allow people with visual impairments to develop an autonomous and self-sufficient mobility — *Orientation* denotes the knowledge of where you are and want to go next, while *Mobility* implies the ability to roam around safely and efficiently. Its main tools in use today are the "White Cane", specially trained "Guide Dogs" as a companion, and *Echolocation*, i.e., the creation of clicking noises that are reflected by one's surroundings, a skill that is much harder to acquire than "White Cane" usage. The cane itself has also been referred to as "Long Cane" or "Probing Cane," but was originally named after its inventor, the "Hoover Cane."

In 1974, O&M was brought to Germany by Béatrice and Jochen Fischer as well as Pamela and Dennis Cory, all working for "blista" ("Hochschulbücherei, Studienanstalt und Beratungsstelle für blinde Akademiker e.V.," founded 1916 in Marburg), becoming one of the first O&M professional trainers in Germany. To this day, blista remains the sole accredited institution involved in educating professional rehabilitation trainers for people with visual impairments.

Recently released data by the *World Health Organization*[1] (WHO) shows that a large number of people worldwide are affected by severe visual impairments and blindness today. It is approximated that 1.3 billion people are influenced by some form of vision impairment, of which 80% is considered avoidable, *e.g.*, refractive errors using glasses and cataracts requiring surgery. This leaves 36 million blind people and 217 million with moderate to severe visual impairments, with the majority of affected people being over the age of 50 (Bourne et al., 2017). This has become a major problem for low-income as well as high-income countries and while the specific reasons are not fully discovered yet, diseases were recently identified as a contributing factor. As some countries do not provide any government benefits, people there are neglected the most, adding to the fact that visual impairments and blindness are legally defined differently everywhere. While the number of people with visual impairments has stagnated since 2010[2], due to the increase in over-aging of the population in various countries, age-related eye diseases are currently expected to rise again.

The German Federation of the Blind and Partially Sighted ("Deutscher Blinden- und Sehbehindertenverband," DBSV) estimated there were 150.000 blind and 500.000 visually impaired people in Germany in 1990[3]. However, these are estimates based on historical numbers taken from the "German Democratic Republic" (GDR), extrapolated to all of Germany. Newer statistics are not available, due to not being collected in Germany. Resnikoff et al. (2004) examined Denmark, Finland, Great Britain, Ireland, Iceland, Italy, and the Netherlands and found an increase of 80% between 1990 and 2002, mostly due to an ageing population. This suggests there were *circa* 1.2 Million affected people in Germany in 2002. Only *Bavaria*, a German federal state, published numbers for people that applied for government benefits in 2013 (14655 persons): 64.2% were over age 65, 42.1% over age 80 and 58,3% were female (higher life expectancy in Germany), supporting the ageing population theory.

[1] https://www.who.int/blindness
[2] https://www.who.int/blindness/publications/globaldata/
[3] https://www.dbsv.org/zahlen-fakten.html

1.1 Daily Challenges for People with Visual Impairments

Being affected by severe visual impairments can create a high mental burden and cognitive load for everyday tasks, often in ways hard to notice and comprehend by sighted persons. Especially small daily living issues can easily become cumbersome, *e.g.*, tracking and finding objects or moving around at home. All activities and decisions related to color are also much harder or even impossible for affected persons, *e.g.*, to select which clothes to wear in the morning, whether these are dirty, spilled, stained, or can simply be worn again, sorting them for washing, as well as many more simple day to day tasks (*cf*. Bigham et al. (2010)).

Eating food is often considered a mentally draining activity as it requires fine motoric skills, *e.g.*, slicing, cutting and picking up of small pieces, as well as tracking pieces of food around the plate. Drinking beverages can easily become just as straining, and sometimes entertaining, if they have to be poured from the bottle: First the correct bottle needs to be located on the table, shelf or the fridge, and then one must carefully pour the beverage in order to not spill it.

Shopping for articles of many different flavors is just as cumbersome: People with visual impairments can hardly differentiate products of similar shape, *e.g.*, different drinking bottles or different flavors of potato chips, nor can they get an overview of what is available at the store in the first place, *e.g.*, standing in front of the fruit and vegetables section. In many contexts it would generally not be considered acceptable to touch everything until the desired food object was found.

Whilst there exist some highly specialized assistive devices to alleviate some of these issues, *e.g.*, the "EinkaufsFuchs"[4] or specialized barcode scanner smartphone applications like "Digit-Eyes"[5], these still require the user to first locate the barcode, which can be cumbersome for items of different shape and structure, as well as the device or application to have the barcode in its database of known products. However, even using such an easily indispensable everyday aid, users will not be informed about special offers, new in store offers, and expired or spoiled items. It also remains completely infeasible to just rummage around through a section of items and look for items of interest, neither potato chips, clothing, collectables, nor LP records, a behavior that is otherwise an essential part of the shopper's experience.

[4] http://www.synphon.de

[5] https://www.digit-eyes.com

Finally, social consequences can also become an issue. It is impossible to pick up on important social cues in conversations, like eye-rolling or mouth-twisting, or even just knowing — in a group of people — when it is one's turn to speak up or whether one was the addressee of a question, a situation that is for example often encountered in work interviews or meetings. Furthermore, participation in social activities becomes much harder, it often only starts with getting to a meeting point self-determined and independently, as such seemingly simple tasks require more time and careful planning — often help by others is needed.

Today's navigation is becoming increasingly complex, as road network density increased due to packed city layouts. On the other hand, modern navigation systems have become ubiquitous — almost any modern smartphone can run the Google Maps application, open it in a browser[6], or run other navigation software — so everybody can literally carry this capability around within their own pockets. However, only the last years have seen an improvement in routing for systems available to the public: Google Maps added a pedestrian routing mode in 2015, providing an improvement over road-based routing that was already commonly found in cars and provided the base for all existing routing approaches so far. Some other routing providers have followed suit in the meantime. Sadly, none of these existing systems address special requirements, *e.g.*, cerebral palsy, walking aids, or people with visual impairments.

However, navigation for these persons only starts with finding their own home or work address, it is a general issue. While most navigation systems will route towards the desired street, and increasingly up to the specific address, for example the roadside to choose for walking on is rarely considered. Moreover, provided navigation information is usually rather *coarse*: "Turn right in 200 meters." or "Follow this road for 100 meters." is missing essential information to people with visual impairments. Not only dynamic obstacles, *i.e.*, people, cars, or bicycles, are completely ignored, but also static objects like trash cans, lamp posts, overhanging branches, low road-signs, a truck's loading ramp or construction sites are also not considered at all.

This is where Orientation & Mobility training provides great assistance, as it teaches specially designed techniques that deal with such situations and also exercises often taken routes. It allows trainees to get used to actually roam around independently and self-sufficiently — and increase their level of confidence while doing so. One-on-

[6] https://google.com/maps

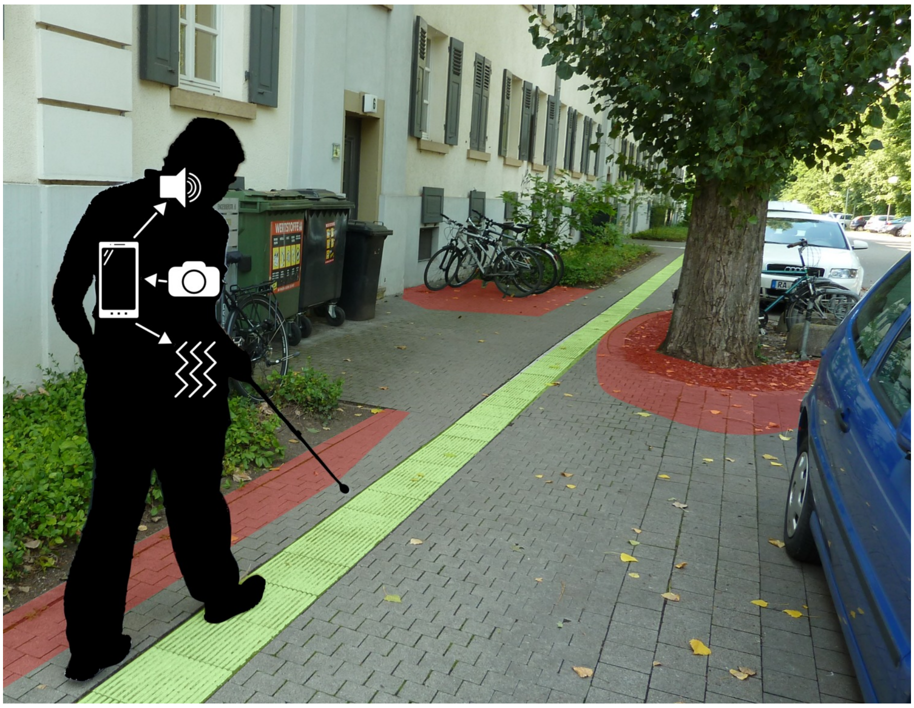


Figure 1.1: An exemplary urban navigation scenario that shows a (silhouetted) pedestrian use a White Cane, shorelining along a tactile paving surface alongside a building. Additional relevant information about the surroundings — of the kind that our assistive system provides, such as accessible section, obstacles or shorelines — helps in improving spatial awareness.

one sessions with O&M trainers are necessary to develop essential skills and properly perceive and understand the surroundings. However, while personal O&M training provides the greatest benefits, it is also very costly and time consuming, and a very high level of autonomy is required to safely navigate unknown areas in the end.

Rarely, tactile pavings, invented by Seiichi Miyake in 1965 to assist a blind friend, provide an as easy and safe path to follow along as in figure 1.1. Usually, people with visual impairments rely on O&M's *shorelining* technique, *i.e.*, the White Cane trails a physically distinctive object, *e.g.*, a building's facade (the *inner* shoreline) or the junction between the sidewalk and curblines (the *outer* shoreline). However, these shorelines are often discontinuous, *e.g.*, when crossing a street or a driveway. Also, most pedestrian crossings provide low accessibility by make and design, and are, if possible, avoided by people with visual impairments (Matthews et al., 2014).

In summary, provided navigation information must not contradict or counteract the intricate Orientation&Mobility techniques learned and memorized by people with visual impairments, but instead consider and integrate these special requirements.

1.2 Computer Vision: A Potential Savior

Computer vision presents itself as a suitable and somewhat obvious candidate to alleviate the effects of visual impairments, due to its core value of "Teaching computers to see!" — a phrase often used to describe it to outsiders. However, computer vision itself is not enough in order to be truly useful, as it needs further understanding of user requirements, their specific needs and capabilities. While modern approaches are capable of identifying, locating, segmenting — marking pixel-wise in an image — and tracking all kinds of objects, they cannot understand what would be considered useful by people with visual impairments in the first place. Thus there exists a very high potential to overload the user with unnecessary details, instead of only providing situation aware relevant information.

Classical computer vision approaches involve hand-crafted features and decision systems, *e.g.*, Scale Invariant Feature Transform (SIFT) (Lowe, 1999), Histograms of Oriented Gradients (HOG) (Dalal and Triggs, 2005), Local Binary Patterns (LBP) (Ojala et al., 2002) or Support Vector Machines (SVM) (Cortes and Vapnik, 1995). These were usually single class object detectors, fine tuned for a specific use case — a specialized creation, often cumbersome and time-consuming. One major benefit is that they do not require much training data, as the developer provides part of the reasoning and integrates assumptions based on external knowledge. Furthermore, it is feasible to reason about their limitations, while failures can be expected and counteracted beforehand.

Recently, deep learning has significantly changed all of this, as it allows *end-to-end* training of Neural Networks, *i.e.*, systems learn the necessary features and classifications fully automatic, all based on data alone (Krizhevsky et al., 2012). It is heavily Graphical Processing Unit (GPU)-based, a computing unit that is capable of running a single instruction on multiple data (SIMD), *e.g.*, to run an addition or multiplication on a large vector or matrix, on all their elements at once. It is widely used nowadays and often beats classical approaches in almost all scenarios it has been tried for, *e.g.*, multi object detectors, segmentation, depth estimation, and many others. One of the major drawbacks is that it requires lots of data in order to be useful, because it needs to properly generalize from its training data to yet unseen instances — a task that had been offloaded to the developer before. This is due to the fact that machine learning usually fits itself to known data as closely as possible: Interpolation becomes easy, while correct extrapolation to unknown data is much harder to achieve. Furthermore, guarantees about functionality under certain circumstances are hardly possible — an especially important fact when human lives depend on it. Thus, a better understanding of its inner workings and its major reliability factors is still required. However, deep learning has recently proven itself to be especially relevant for driver assistance systems and autonomous driving, both important research fields that can, albeit to a limited extent, greatly benefit people with visual impairments as well.

Traditional systems were often limited in their mobility and thus also in their usability for people with visual impairments as navigation and guidance aids. However, with the advent of smartphones and tablets, ever more processing power as well as connectivity is now available — that is also highly mobile. Today, we additionally have much longer lasting batteries due to improvements in power consumption and battery technology, but also further technologies available as well, *e.g.*, bone-induction headphones, Global Navigation Satellite System (GNSS), and many more. Furthermore, mobile GPUs are recently coming to smartphones and tablets as well, allowing the use of deep learning approaches, even in mobile scenarios. There is a general increase in the number and the availability of “off-the-shelf” hardware, making systems much more affordable and easier to develop on. Still, personalized mobility so far is limited to GNSS-based routing systems, while it could be digitally enhanced by relevant — user-centric — information about the immediate surroundings.

Especially today’s ubiquitous smartphones are becoming relevant in everybody’s daily lives, for sighted as well as for people with visual impairments. Nevertheless, smartphones present their own, very unique, challenges for people with visual impairments, due to their lack of tangible feedback when using the touchscreen. Thus, many affected people, i.e., people with visual or motor disabilities, carry various types of specific use case devices, as observed by Kane et al. (2009). These devices also present accessibility challenges or consider only a very specific scenario and thus provide only limited assistance or require heavy user customization to become truly useful. Meanwhile, accessibility has much improved: Apple iOS’ “Voice-Over” and Google Android’s “TalkBack” have gained much popularity (Csapó et al., 2015), as well as few specialized applications, *e.g.*, “BlindSquare”[7].

[7] https://www.blindsquare.com

1.3 Contributions & Outline

The main objective of this thesis is to reduce the currently existing *spatial* and *systemic* gap between Orientation&Mobility's "White Cane," as well as navigation and routing approaches based on GNSS:

Spatial Gap There exists a huge void and complete disconnect between the close-up range that can be perceived using the White Cane and high-level routing information provided by a navigation system.

Systemic Gap This void has systemic origins, as the GNSS does not consider dynamic, *i.e.*, moving, obstacles and is thus not capable to fully assess the local conditions, while O&M's White Cane techniques are disconnected from the additional high-level information about the surroundings and upcoming route that a navigation system provides.

Such gaps largely exist because of essential missing information about routing in the immediate vicinity of the user. We use body mounted cameras and computer vision approaches to close these gaps, *e.g.*, how to actually get to the "Right turn in 200 meters!" without walking into dangerous obstacles, while still walking the shortest and most direct route towards the desired destination. Furthermore, already acquired Orientation&Mobility skills, *i.e.*, White Cane techniques, must be accounted for and our provided information must not contradict such skills, but instead work hand in hand to provide a safer route and less stressful experience. Generally speaking, we virtually elongate the White Cane, significantly increase its working range and allow an early notification about upcoming obstacles. This allows the user to improve spatial awareness for the immediate surroundings, especially in unknown areas.

This thesis is structured as follows:

Chapter 2: Background & Related Work. This chapter first presents an overview of related literature for general navigation and guidance advice as well as general issues and misconceptions about assistive systems for people with visual impairments in section 2.1. It is followed by orientation related literature examined in section 2.2 and separated by their distinctive technological features, as well as related works for relevant aerial imagery processing (Section 2.2.1) and impairment aware routing (Section 2.2.2). Afterwards, mobility related works are discussed in section 2.3

and especially relevant literature for our accessible section detection (Section 2.3.1), shoreline detection and tracking (Section 2.3.2), as well as specific works for pedestrian crossings (Section 2.3.3). We then present a small — very selective — overview of user studies in section 2.4 that have influenced our own designs and conclude this chapter in section 2.5 by listing our identified key concepts and takeaways.

Chapter 3: Orientation. In this chapter we take an in-depth look at relevant high-level information needed for special requirement centered navigation and routing. We propose situation specific assistive systems and improve available data for impairment aware routing processes. Our contributions in this area are:

- We analyze aerial imagery to improve the availability of relevant geospatial databases that are required for impairment aware routing algorithms by searching for zebra crossings in urban areas (Koester et al., 2016). (Section 3.1)

- Furthermore, we propose an impairment aware routing approach for people with visual impairments (Koester et al., 2017), which integrates a shoreline level of detail into generated routes. Through this requirement aware routing, dangerous situations, *e.g.*, unsafe road crossings and especially uncontrolled crossings, are prevented if possible or made safer by integrating nearby available accessible crossing locations instead. Additionally, such fine-grained routing — shorelines are a major contributor to safety and orientation — up to the next GNSS-based navigation point potentially improves spatial awareness of one's surroundings, as shorelines and Accessible Pedestrian Signals (APS) are incorporated into the routing process and relevant information about next steps are made available in advance, which we analyze in a Wizard of Oz experiment. (Section 3.2)

Chapter 4: Mobility. We first analyze basic mobility requirements, identify issues and provide three situation-based assistive systems to alleviate identified shortcomings and assist in these specific mobility tasks. Our guidance contributions are:

- We propose real time mobile accessible section, and thus also obstacle aware, detection to assist in safe and self-dependent mobility in known and unknown terrain. This prevents bumping into obstacles that cannot be perceived using the White Cane and virtually elongates the cane, so that obstacles can be better noticed from afar. (Section 4.1)

- We also propose a real time capable approach to detect and precisely locate inner shorelines, *i.e.*, the junction of an accessible section and a vertical surface such as natural features that are used for shorelining with the White Cane (Koester et al., 2018). This system is especially useful in urban areas, where the shoreline, *e.g.*, a building's facade, is interrupted by a driveway or an intersection, as the next relevant shoreline can be presented to the user in advance. (Section 4.2)

- Finally, to assist in pedestrian crossings, we provide additional precise location information, *i.e.*, zebra crossings, pedestrian walk lights, pushbuttons, and pavement markings. Furthermore, the state of a traffic light is detected and communicated to the user. We use a deep learning approach, optimize, and evaluate it, and test it as part of the "TERRAIN" project in a user study (Koester et al., 2019). (Section 4.3)

Chapter 5: User Studies. Aside from quantitatively evaluating our already mentioned contributions individually, we also test our assistive system qualitatively, *i.e.*, in isolation as well as in useful combinations, as part of a larger user study alongside other aspects. Therefore, we integrate our computer vision algorithms into a real time capable mobile prototype (Section 5.1) used by study participants that had to provide sufficient processing power and battery lifetime, enabling the feasibility of these user studies in the first place. Finally, we discuss the results of the TERRAIN user study in section 5.2.

Chapter 6: Conclusion. We summarize the thesis and outline our main contributions made in the field of Orientation&Mobility assistance for people with visual impairments in section 6.1. We then conclude our thesis suggesting future research directions — proposing open questions in section 6.2 that have the potential to further improve our created assistive systems as well as systems for people with visual impairments in general.

A full list of made publications can be found within the appendix, including contributions to works not further discussed in this thesis.

Chapter 2

Background & Related Work

> "If I have seen further it is by standing on ye sholders of Giants."
>
> – Isaac Newton

No research effort is conducted in a vacuum, as the above quote by Isaac Newton implies. While there exists an exuberance of proposed systems for Orientation & Mobility, we try to provide a possibly complete overview of those created within the last two decades, but might have still missed a few, so this chapter provides an overview of relevant background and related works. It also discusses the state of the art for the various fields necessary to fully understand this work and its specific motivations. Furthermore, it hints at works that were highly influential to our own ideas and specific designs, as well as differentiates our own work from those mentioned here.

Although created systems often incorporate more than a single aspect, we have tried to categorize them by their most important contribution and sensory capabilities. The two major objectives lie within this thesis' focus of orientation and mobility. But first, we start with an explanation of some common issues and misconceptions when it comes to O&M for people with visual impairments in section 2.1. We then provide an overview of proposed orientation and navigation approaches in section 2.2, and refine these for aerial imagery processing (Section 2.2.1) and impairment aware routing (Section 2.2.2). Afterwards, we discuss several systems for mobility and guidance in section 2.3, especially those related to accessible section and obstacle detection (Section 2.3.1), shoreline detection (Section 2.3.2), as well as two different kinds of pedestrian crossings (Section 2.3.3). We further separate these Orientation&Mobility systems first into works that have provided us with general insights and then based on their used sensors, *i.e.*, into sonar, Time-of-Flight (TOF), computer vision, GNSS/Inertial Measurement Unit (IMU) and wireless technology-based systems, where applicable. We proceed to discuss selected user studies, conducted as part of the described Orientation&Mobility related works in section 2.4, specifically those that have somehow inspired or influenced our own decisions. To conclude this chapter, we present our key concepts and takeaways in section 2.5.

Own Contributions. At the end of every section that is closely related to some of our own work, we included a short paragraph that provides a gist of our contribution and references the relevant detail chapters. Such paragraphs exist for: aerial imagery mining in section 2.2.1 (page 18), impairment-aware routing in section 2.2.2 (page 21), accessible section and obstacle detection in section 2.3.1 (page 26), shoreline location and tracking in section 2.3.2 (page 27), and finally, pedestrian crossing assistance in section 2.3.3 (page 33).

2.1 Common Issues & Misconceptions

This section lists some research works that point us to important issues often forgotten when designing assistive systems. It also describes a few misconceptions about people with visual impairments that we identified as very relevant to our own work.

Surprisingly, it is very hard for most, if not all, humans to walk in a straight line or follow a straight direction for a prolonged period of time if there are no fixed reference points available, *e.g.*, the sun or distant objects. Sooner or later, blindfolded people will start to veer of their path and walk in rather small circles, with diameters of less than 20 meters (Souman et al., 2009), with no preference of a specific direction. People with visual impairments are no exception to this, as the authors state, since such veering is most likely the result of noise accumulating in the *sensorimotor* system, similar to an IMU, when there is no external reference to periodically recalibrate it.

Brady et al. (2013) suggested that most challenges faced day to day by people with visual impairments are not well understood. They developed the “VizWiz Social” application for iOS, based on “VizWiz” (Bigham et al., 2010), where 5.329 users asked 40.728 questions about images they took with their mobile phones, which were then answered by Amazon Mechanical Turk workers. Question types ranged from identification of objects by name or type to read image text of any kind, description of visual or physical properties of various objects as well as other questions, that sometimes could not be answered properly. While there were no questions about Orientation & Mobility — this is possibly due to its minimum latency selectable by users of "Within a minute" — this work still provides a good sense of the daily challenges faced by people with visual impairments (*cf*. Section 1.1), which are often different than what would be assumed by sighted people.

Williams et al. (2013) analyzed how people with visual impairments integrate technology into Orientation & Mobility techniques to support navigation and the complex as well as personal choices they make when using such tools. They also reported that most users rely heavily on O&M training, rate it as excellent and very helpful and use the techniques exactly as they were taught. Only a few have modified or abandoned some learned techniques due to highly individual reasons, *e.g.*, only moderate vision impairment. It was also stated that guide dogs and White Cane users behave significantly different in navigation tasks — a guide dog immediately walks towards a target whereas a cane user seeks outs a shoreline first. This and many more differences have to be accounted for by an O&M assistance system. They then further investigated the misconceptions of sighted people when acting as a sighted guide w.r.t. providing feedback to people with visual impairments, as well as the misunderstandings and frustrations this often causes for the latter group and provided specific guidelines on how to improve these in the future (Williams et al., 2014). One of many observations was that the inclusion of shorelines as well as their differentiation to actual obstacles should be made clear to the sighted guide beforehand. It was also observed that relevant hazards were not considered, due to their obvious nature to sighted persons, which eventually caused the provided assistance to be incorrect and also to interfere with O&M techniques.

Finally, a recent survey by Elmannai and Elleithy (2017) gives a comprehensive overview of assistive devices for people with visual impairments and discusses their benefits and limitations in great detail. It notes that while there already exist some specialized systems, none of these were 100% satisfactory w.r.t. to essential features identified by the authors. It concludes with a set of essential aspects to be considered for future assistive systems: performance, wireless connectivity, reliability, simplicity, wearability and economical availability.

2.2 Orientation — Navigation

Similar to other surveys in this area, Csapó et al. (2015) notice about assistive navigation applications on mobile platforms that these applications are largely inaccessible and fail to adapt to special needs, or often require expensive specialized hardware as well. While there exist many applications for navigation, *e.g.*, Garmin's On the Road, Google Maps, TomTom's GO Mobile or Waze, to only name a few of them, as well as of course their dedicated units that do nothing but navigation, almost none of these are accessible to people with visual impairments, which thus have had to rely on specialized devices in the past, *i.e.*, the Trekker Breeze. Therefore, this section does not include such applications, but instead focusses on research systems that have been proposed in the past.

GPS/Compass/IMU-Based Systems

Poor Global Positioning System (GPS) quality (*cf*. Section 2.2.2) was a huge issue for the first prototype systems, such as the one proposed by Loomis et al. (1994). They had to select specific times of the day where satellite availability was especially high, *i.e.*, 6 or 8 satellites were within reach, and also ensure that shadowing due to buildings or dense foliage were not an issue. Even under optimal conditions they had to use a Differential Global Positioning System (DGPS) — a system where an additional, fixed position, receiver propagates its own GPS error to other nearby devices.

Ross and Blasch (2000) analyzed three different navigation interfaces, a stereophonic sonic guide, speech output and a shoulder-tapping system, *i.e.*, a small array of small contact speakers that each produced a light thump on the person's back. They use it to prevent veering off during a large crossing and found that the tapping interface provided the best results, and would produce even better results if complemented by a speech system for general navigation instructions.

Almost at the same time, Helal et al. (2001) proposed a similar system that also uses DGPS for outdoor positioning and in addition ultrasound for indoor navigation. They generate step-by-step walking guidance information for indoor environments, however, this has to be done manually beforehand using a specially created Geographic

Information System (GIS) database. Furthermore, a wireless connection is used to serve the GIS database to the user, which severely limits the system's working range.

GPS and dead reckoning-based on a compass are proposed by Pressl and Wieser (2006) in order to create their PONTES system, consisting of a navigational environment model, fast routing algorithms to generate lists of maneuvers, positioning tools, map matching algorithms and adequate guidance instructions. In a follow-up project, Mayerhofer et al. (2008) created ODILIA to solve the remaining problems of their first system, *e.g.*, include the use of public transport systems and to optimize the generation of the path network used for their navigation algorithms, as well as a sonar-based aerial obstacle detection. While they provided a rather complete prototype system, they note that the integration and overall size need to be improved as well as large maps created that contain the additional data they rely on for their routing.

Sánchez and de la Torre (2010) utilize a Microsoft pocketPC with external GPS sensors and create an audio only navigation app for it. They note that important information to truly allow autonomous navigation of people with visual impairments is still missing, *e.g.*, obstacles in the path, curbs, roadwork construction sites or storefronts. Finally, Katz et al. (2012) propose a different interface, *i.e.*, spatial 3D audio, to communicate navigation information to the user in their NAVIG system (*cf*. Kammoun et al. (2010)).

While we also use GNSS for impairment aware routing, we identify the requirement to combine it with computer vision, *e.g.*, to detect obstacles or integrate shorelines. We also decided to use a combination of haptic and acoustic output for our own systems, as we learned from above examples that this seems to be the best option.

BLE/RFID/Wi-Fi-Based Systems

Ross and Lightman (2005) proposed to distribute "digital crumbs" around buildings and indoor public spaces. Tiny, solar-powered signs that include their own data, processing chips and a speaker are activated through a body-worn badge that identifies to the system using infrared as well as the Zigbee protocol, a low power alternative to early Bluetooth versions.

Meanwhile, Tatsumi et al. (2006) suggested to create a network of Radio-Frequency Identification (RFID) tags to communicate landmarks in the immediate environment, by simply attaching unobtrusive, though not cheap, active RFID tags on demand, not just to static landmarks, but also to moving obstacles. The authors then analyzed various location and tracking of objects scenarios using small mobile robots, in an attempt to create and also continuously update a map that could later be used for people with visual impairments. Na (2006) also suggested a RFID-based system for indoor positioning. Their proposal goes even further than those of other RFID-based systems: they intended to put a unique passive RFID tag under each floor tile, thus creating a very dense network of tags that allow very precise positioning within equipped buildings.

Instead of pure RFID connections, Bohonos et al. (2008) also proposed to rely on other wireless communication technologies, *i.e.*, Bluetooth. They intended to use it for pedestrian crossings, by fitting such technologies onto traffic lights as well as city buses in order to communicate up-to-date information to pedestrians and passengers. Fernandes et al. (2011) suggest a very similar system, integrated into a White Cane in combination with a smartphone, that relies on active RFID tags as well, in order to communicate traffic light state, and passive RFID tags integrated into the pedestrian crossing pavement itself. Finally, Mehigan and Pitt (2012) used Bluetooth and a wireless IMU unit to improve spatial orientation and "Dead Reckoning" on their campus and tested this system on a specific route with some success.

Of all proposed research systems, NavCog has probably become the most noticeable of these efforts in recent years (Sato et al., 2017). Its authors also aspire to significantly improve indoor navigation and use similar technologies, *e.g.*, Bluetooth Low Energy (BLE) and IMUs to provide precise indoor localization. After a building has been equipped with a sufficient amount of active BLE beacons and a digital model created, they can provide very fine-grained guidance information to a user in their smartphone-based application. Additionally, depending on the created level of information in their spatial map, they can also provide a huge amount of Points of Interest (POI) related information, based on the user's desires and preferences. They tested their system in a 21.000 m^2 shopping mall with very convincing and promising results.

While all of the above mentioned wireless technologies have the potential to provide fast, mostly exact and up-to-date information of many different kinds, *e.g.*, pedestrian walk light states, incoming buses or tram lines, intersection layouts, shopping store locations, low-level guidance, *etc.*, without much room for errors due to the stability of the chosen technologies, they require a quite laborious and often expensive infrastructure to be installed on every location that should become more accessible and are thus not feasible for widespread distribution. This is exactly why computer vision provides strong benefits for these and similar use cases, as it does not rely on *prior* installment of dedicated infrastructure, but instead tries to accumulate the same level of information using methods that are applicable everywhere.

Computer Vision-Based Systems

For tasks that focus on general navigation of people with visual impairments, computer vision-based techniques have only been recently proposed. Devices like Google Glass were used by Altwaijry et al. (2014) to localize the user based on GPS first and in a second step perform an image analysis in the cloud to recognize buildings in the image and provide a description from online services.

Leung and Medioni (2014) use a Vuzix Wrap system, *i.e.* a pair of glasses with a built-in stereo camera and an IMU, in order to localize the ground plane and create a trajectory to better track and locate a user in a navigation scenario.

Another vision-based technique is door localization, an often occurring problem for people with visual impairments when using navigation systems, as the last few meters towards the precise location of a door are never accounted for. Tian et al. (2010) detect these using a camera integrated into a pair of sunglasses or a cap with geometric modeling and show a very nice performance, but lack an interface and a user study.

Later, Fiannaca et al. (2014) also used door localization as one of many features to help users traverse large open spaces, to prevent veering off and compare the effectiveness of sonification and text-to-speech output.

Computer vision seems rather unsuited for high level navigation tasks when compared to GNSS or wireless technologies, as the general lack of systems in this domain suggests. Nonetheless, there is still great potential for computer vision systems for orientation issues that do not necessarily require such a high level knowledge. In the rest of this section we provide an overview of systems built towards a few specific scenarios where computer vision is successfully used to provide such low-level orientation.

2.2.1 Aerial Imagery

Aerial imagery has been under investigation for quite some time, often to detect and monitor roads in satellite imagery (Quam, 1978). Special interests such as city planning require a precise segmentation and classification of this kind of imagery in order to provide correct geospatial data (Hinz and Baumgartner, 2003). Máttyus et al. (2015) much later propose a data driven approach from OpenStreetMap (OSM) data and aerial imagery for road surface detection and segmentation that requires comparatively little amount of data and generalizes well.

Section 2.3.3 provides an overview of systems created to detect zebra crossings from street-level positions. While none of these algorithmic approaches have been created for detection in aerial imagery, some of them could eventually be made to work after some tweaking, since a zebra crossing has such unique and clearly visible features from a street level perspective as well as in aerial imagery.

However, it has not been until very recently that zebra crossing detection from aerial imagery was first proposed by Ahmetovic et al. (2015). They suggest to download aerial imagery from Google Maps, limited to partitions that contain road surface, which is calculated using the distance of a partition's center and the next road, information provided by the Google Maps Javascript API. After processing these partitions, they generate a list of zebra crossing candidates that are then verified against Google Street View panoramas in a second step. This two-stage approach depends purely on a line search algorithm, for both the initial detection stage in aerial imagery, as well as the candidate verification stage in the panoramas. The authors evaluate their system on a 1.6km^2 rectangle in an urban San Francisco area and achieve very good results.

Our Contribution. We improve the state of the art for zebra crossing detection in aerial imagery using a purely data driven approach (*cf*. Section 3.1) compared to other proposed methods that were all hand-engineered and hand-tuned as well. Furthermore, we are able to do so with less data, *i.e.*, we do not require to verify our candidates using Google Street View and rely only on aerial imagery. Finally, we improve available geospatial databases and also show that our approach can even be used across countries, comparing it to state of the art.

2.2.2 Routing

In 2007 an explosion of interest in geographic information created new communities dedicated to Volunteered Geographical Information (VGI), a term coined by Goodchild (2007), who first noticed this particular phenomenon and attributed it to various factors. The author considered self-promotion and personal satisfaction as driving factors for people that contribute to projects such as OSM or Wikimapia[1], but also the wish to have an assertible data source for geospatial data that is not controlled by a private entity, *i.e.*, Google Maps. Haklay and Weber (2008) also noted that volunteer engagement increases when spotted errors can easily be corrected. However, they have also observed an increasing inequality in the amount of contributions by individual participants as well the greatly differing quality of data in various regions.

Especially for people with visual impairments a huge range of relevant information can be gained from such VGI and ideally be integrated into routing systems. Hara et al. (2013a,b) rely on crowdsourced data in order to identify street level accessibility issues as well as to improve bus stop accessibility. They use Google Street View images and untrained Amazon Mechanical Turk workers to asses sidewalk accessibility issues and achieve very good results, demonstrating the scalability of their work. Sadly, the authors state that created labels do not provide sufficient quality for the development of computer vision algorithms, much less for the training of modern deep learning approaches.

Impairment Aware Routing

Before May 1^{st}, 2000, selective availability of Global Positioning System (GPS) signals was only available for the United States military. The general public could resolve a GPS receiver's positions only within 100 meters. After the change introduced by then President Bill Clinton, acquisition accuracy increased to between 6 and 10 meters under normal conditions (Haklay and Weber, 2008). Before, it was practically impossible to create routing systems for urban areas. It did not take long after this switch and very early prototypes for GPS-based navigation system for people with visual impairments were suggested, *e.g.*, Guillet et al. (2002).

[1] https://wikimapia.org

Völkel and Weber (2008) developed a collaborative system that generates routes not just for people with visual impairments, but for a multitude of mobility impairments. They allow such user groups to share their created routes as well as to adapt them to their individual needs, even keeping user's privacy in mind while still benefitting from the sharing process.

Another adapted routing algorithm was proposed by Kammoun et al. (2010). Using a manually created geospatial database, developed after interviews of people with visual impairments and O&M instructors, they search for a highly adaptable route according to a user's specific needs. The resulting system was then integrated into NAVIG, where the computer vision part was used to improve user localization using visual landmarks.

A different approach was offered by Rice et al. (2012). Like many other routing systems, the authors rely on VGI to update a database of messages regarding the local area, the gazetteer — a geographical dictionary. Key aspect of their work is the gazetteer's update process that semi-automatically matches geographically referenced VGI updates with local place names. Using open source technology, they publish automatic alerts and map updates to their users, allowing these to adapt their routes before leaving for their destination.

While Guy and Truong (2012) used annotated Google Street View images to provide users with relevant information at pedestrian crossings, Carrasco et al. (2014) have developed a navigation system that also covers pre-planning and binaural audio-based guiding techniques on a mobile device. Using GNSS and an IMU, the authors provide very detailed heading instructions of planned tracks over open bone conduction headphones and demonstrated the binaural guiding functionality with four end users.

Andreev et al. (2015) proposed realistic pedestrian route planning by analyzing specific requirements and differences of pedestrians compared to other routing entities, *i.e.*, cars. First, pedestrians can utilize large traversable areas more freely, usually in order to minimize their walking distance, since they are not bound to pre-defined road networks. Second, pedestrians prefer longer but safer routes and preferably utilize pedestrian bridges and underground connections. Although it is currently infeasible for people with visual impairments to traverse large open spaces, due to missing reference points, their routing approach still provides a great value, since it allows a high level of personalization.

Schmitz and Ertl (2014) instead generated extended routing networks using map content transformations. Given additional parameters of roads, *e.g.*, the number of traffic lanes, the width of road surface or — ideally — the existence of sidewalks, they create very precise pedestrian walkway networks (*cf*. Schmitz (2015)). However, these transformed maps merely represent a specific's model view of the world and often do not precisely match with underlying conditions found in the real world. Thus, although they exhibit the benefit to provide very fine-grained routing, *i.e.*, if existing GNSS accuracy allowed this, they have to be used with much care.

Spatial awareness, *i.e.*, having a spatial map of one's surroundings or a planned route available beforehand, has been identified by multiple works as a contributing factor for a safe and natural navigation, *e.g.*, Schneider and Strothotte (2000) that used constructive exploration where a force feedback system is used to interactively discover and learn a route in virtual reality, or Yatani et al. (2012), who also investigated tactile feedback for map discovery, but on a mobile device equipped with multiple vibration motors.

Other approaches have dealt with POIs and how to transfer such information to a person with visual impairments, *e.g.*, Yang et al. (2011) and Blum et al. (2013). Both found in their conducted user studies that such POI-based systems have the potential to aid in spatial awareness, if the provided information is relevant for the desired navigation context, *i.e.*, objects that are in a user's immediate vicinity and read out as sorted by user defined categories. However, Guerreiro et al. (2017) also suggested that route pre-planning and usage of POI are most useful in different interaction contexts, *e.g.*, known versus unknown areas, and therefore further research into these specific scenarios is required.

Our Contribution. In our work we address the specific needs of people with visual impairments and propose an original — shoreline level of detail-based — routing approach (*cf*. Section 3.2). We base our routing on a specific O&M technique, *i.e.*, shorelining, and integrate available shorelines that we extract from relevant OSM data. Additionally, our routing algorithm has the capability to create the safest route based on user preferences, *e.g.*, to avoid certain crossing types if possible. Finally, our approach helps in the creation of a spatial map — it potentially improves spatial awareness of the surrounding's POIs that are relevant for O&M techniques.

2.3 Mobility — Guidance

First computer vision-based systems to assist in mobility used very simple vision techniques, due to the limited available processing power on mobile devices, such as simple color markers (Chan et al., 2007). Manduchi et al. (2008) use such markers for indoor navigation and analyze various search strategies as well as discuss their general usability for this task (Manduchi et al., 2010). While such systems work nicely when a building has already been outfitted with such markers, they do not scale well to all the places people with visual impairments might want to rely on such a marker-based technique.

Manduchi (2012) also analyzed whether computer vision techniques are a feasible technology for people with visual impairments in the first place. He identifies several key issues in such technologies, *e.g.*, the limited field of view of mobile phone cameras requires the user to actively explore the environment, while several participants at the same would have preferred for the camera to be attached to their garments, which would hinder active exploration even more. Furthermore, the exact locations of targets that users should scan for have to be chosen very carefully in order to yield acceptable results.

However, Manduchi and Coughlan (2014) later acknowledge that just increasing the field of view might hurt performance as well, due to providing less precise location information when the field of view grows beyond a certain point. Nonetheless, a higher frame rate generally improves performance. The authors conclude that a wide field of view is preferable to a narrower one when spatial resolution as well as precise camera pin-pointing are not required.

A totally different and non-computer vision-based approach was proposed by Shiose et al. (2004). The authors propose to teach the perception of crossability using acoustic 3D environments, a way that would allow people with visual impairments to improve their hearing perception without the dangers of actual urban traffic. Similarly, Sánchez et al. (2010) suggested to use video games as another teaching tool to improve O&M skills. Finally, inspired by recent advances in autonomous driving, Sucu and Folmer (2014) suggest to improve self-determined O&M of people with visual impairments by allowing them to drive a car independently. In a simulator, users were required to complete a test track while they receive haptic cues on the steering wheel about the direction of the road relative to their car's movement. However, while such efforts show only limited use on real roads, they still allow people with visual impairments to participate in competitive computer game sessions with their friends.

A very modern technology-based approach to guidance has also been proposed by Avila Soto et al. (2017). The authors analyze the feasibility of using small drones — unmanned aerial vehicles, *i.e.*, quadcopters — to guide people with visual impairments, either with or without a leash and observe an improvement over pure audio-based navigation. However, they also note some issues w.r.t. social acceptability of using drones as assistive mobility systems (Avila Soto and Funk, 2018), especially with regards to the safety of other passengers by cited bystanders as well as concerns about the personal image by people with visual impairments themselves.

2.3.1 Accessible Section & Obstacle Detection

Dakopoulos and Bourbakis (2010) provide an overview of developed and proposed assistive systems for people with visual impairments between the years 2000 and 2010. While there exist three major categories — Electronic Travel Aids (ETAs), Electronic Orientation Aids (EOAs) and Position Locator Devices (PLDs) — the authors focus on mobile and wearable obstacle detection and avoidance systems. Their results indicate that while most systems provide great value for limited scenarios, they all lack in at least one important aspect. Such major aspects are: *free-hands*, *i.e.*, the ability to use the system in addition to the White Cane, *free-ears*, *i.e.*, environmental awareness should not be limited, *wearable*, *i.e.*, systems can only be useful when they can be actually worn in mobility scenarios, and *simple*, *i.e.*, easy and intuitive to use, as in not requiring any special training. These key aspects have also influenced our own research efforts and are a major design concern for our developed components as well as the overall system.

Sonar-Based Systems

One of the mentioned devices is "CyARM", proposed by Okamoto et al. (2004) as another interactive device to increase a White Cane's sensing capabilities. Here, an ultrasonic sensor gathers distance information and controls the user's arm position, *i.e.*, the distance from its own body through a wire connected to the hip region, to communicate the perceived obstacle distance for the direction the user is pointing towards. Ono et al. (2006) later expand this concept with a flashlight-like attention pointer to be used by a sighted person, that can guide the attention of a person using the system when observing the same object.

Gallo et al. (2010) also enhance the White Cane itself with ultrasonic sensors, to be used as distance measurement and obstacle detection, as well as for aerial obstacles, and use haptic feedback to inform the user about potential issues.

A third sonar-based system, this one specifically for pedestrian crossings, was suggested by Hashino and Yamada (2010). However, it requires *prior* installation of ultrasonic receivers at the other end of a pedestrian crossings in order to be useful, as the user carries the emitter and is positioned via triangulation, thus does not scale easily.

TOF-Based Systems

Many first systems to replace the White Cane were TOF-based, such as the "Teletact" proposed by Bellik and Farcy (2002). Its purpose was to enhance the White Cane's reach, using a laser emitting device, where obstacles would reflect the laser and the device would register this reflection, calculating a distance to the obstacle. Obtained ranges were communicated using musical tones, for different intervals between ten centimeters and thirty meters. Jacquet et al. (2004) then analyzed how to define context-awareness formalism of such a device, in order to model the structure of buildings or name objects.

Zeng et al. (2012) propose a TOF camera and planar tactile display to also enhance the White Cane's capabilities and later improve its design and user interface (Zeng et al., 2017).

Bhowmick et al. (2014) use a Microsoft Kinect camera (infrared-based for depth data) and machine learning, *i.e.*, a Support Vector Machine (SVM), to create a wearable, although indoor only device that also uses image features to detect *a priori* trained objects. Another indoor assistance system by Jafri and Khan (2016) detects obstacles using an infrared sensor as well, built into a Google Tango tablet, and can also use its built-in display to provide details for people with moderate visual impairments.

While these TOF-based systems have the potential to provide great benefits, as they are computationally rather inexpensive due to their sensor types, usability is strictly limited to indoor scenarios. Outdoors, unless at night or on very cloudy days, infrared light transported by sunshine will sadly render these systems mostly useless.

Computer Vision-Based Systems

Possibly the first work to propose only computer vision algorithms for indoor navigation and object identification for people with impairments was Hub et al. (2003). The authors develop a flashlight-like approach that integrates a stereo camera, keyboard control and acoustical output, where the stereo camera is used to recognize an object's color, distance and rough size.

Martínez and Ruiz (2008) used a stereo camera mounted onto a user's shoulder via a messenger bag, but detect aerial obstacles only, such as low hanging branches by simply informing the user whenever there was an object detected within a specific range. Pradeep et al. (2010) presented a head mounted stereo vision camera system that uses visual odometry and feature-based Simultaneous Location and Mapping (SLAM) to build a vicinity map of the environment and to detect ground level obstacles.

The "EYECane," a White Cane enhancement, was proposed by Ju et al. (2009). The authors use a monocular camera attached to a cane that segments the ground. However, this system requires the user to hold the cane still in front of the body while the system is working and thus contradicts the techniques learned in Orientation&Mobility. Similarly, José et al. (2011) proposed to center the user on a path, again an approach that contradicts O&M techniques, but have created an extensive computer vision-based navigation aid that assists in multiple scenarios: detect the path in front of a user and implicitly static obstacles (no path found), zebra crossings, as well as moving obstacles through optical flow, but they use a single earphone or small speaker to communicate this wealth of information.

Another mobile and in this case truly wearable prototype was also proposed by Mattoccia and Macrı' (2015). The system integrates a stereo vision-based obstacle map algorithm integrated into a pair of head worn glasses. A modified Random Sample Consensus (RANSAC) algorithm tries to detect the ground plane in the *v-disparity* domain and sufficiently large deviations in 3 defined Region of Interests (ROIs) are communicated to the user. Schwarze et al. (2015) also suggest a head mounted system. They use a stereo vision camera setup integrated into a bicycle helmet and provide information about obstacles as well as additional scene understanding through acoustic feedback. Owczarek et al. (2016) also use a glasses mounted stereo camera, but do not try to detect obstacles, and instead focus on an intuitive acoustic feedback about reconstructed distances (U-V disparity-based) that allows users to create their own *melody* when walking around.

While such head mounted systems seem to allow a user to *scan* for obstacles using their head movements, this is a technique rarely performed by people with visual impairments, especially rare for those that have been blind since birth, which often also exhibit involuntary head movements as well (Gottlob et al., 1996). Usually, these will turn their head completely independent of their walking direction, for example when they want to improve their attention using their hearing senses towards a specific direction (Wiener et al., 2010). So head worn approaches might as well contradict the actual behaviors exhibited by some people with visual impairments.

A rather recent trend is obstacle detection using smartphones. Sáez et al. (2015) detect aerial obstacles from consumer smartphones with a 3D camera, worn around the neck or hold up in front of the body. Originally, these cameras were put into smartphones to allow people to record videos for their 3D content capable TVs. However, as they can also be used for other purposes, it is praiseworthy that such an active hardware development exists, which improves the integration of mobile components, increases battery lifetime and reduces weight. Thus, it might very soon be feasible to own enough processing power, battery lifetime, sensory capabilities and connectivity in mobile systems to create truly mobile and helpful assistive systems.

Our Contribution. While there already exists an abundance of obstacle detection systems, we propose a depth-based, real time capable, highly efficient and parallelized, as well as highly precise, algorithm to detect the free space in front of a pedestrian (*cf*. Section 4.1). Our approach is based on depth data, because other works have shown that it is the most reliable sensor input, as pure 2D image processing is often not capable to reliably detect deviations from the ground plane. Furthermore, we use a body worn camera, as head mounted systems have also shown some issues with people that are blind since birth and can thus also deal with outdoor scenarios, which TOF-based systems cannot.

2.3.2 Shorelines

An important Orientation&Mobility technique when using the White Cane is shorelining — to follow along *shorelines*, *i.e.*, walls, curbs, hedges or other salient objects, keeping constant contact with the cane or using the arcing left-right touch method (Zimring and Templer, 1983). However, this technique is not considered by any navigational assistance system to date, except for our own method proposed in Koester et al. (2017). It is the first routing approach to acknowledge such special requirements by people with visual impairments using this specific O&M technique and the low level routing considerations that have to be included (*cf*. Section 2.2.2).

Furthermore, computer vision-based shoreline detection for people with visual impairments from a first person's viewpoint has also not been considered at all so far, neither for inner nor for outer shorelines. To the best of our knowledge, the only works that have pursued a similar goal are Coughlan and Shen (2007); Coughlan et al. (2006), who first use a modified *v-disparity* algorithm and later edge-discontinuities in depth maps to detect drop-offs, *i.e.*, curbs, for wheelchair assistance systems.

Ivanchenko et al. (2008a) propose to not detect such drop-offs or obstacles directly for people with visual impairments in a wheelchair. They instead use a stereo vision camera in order to constantly track the used White Cane and the direction it is pointing towards. An alerting sound is then used to inform users about significant deviations from the ground plane within a certain range. Such changes are easy to spot as the stereo vision camera is mounted to the wheelchair and under normal circumstances, the relative position of the ground plane is fixed. As people with visual impairments that not rely on a wheelchair do not provide such a stable, yet moving along, camera position, this technique cannot be directly adapted.

Our Contribution. We propose the first computer vision-based inner shoreline detection for people with visual impairments (*cf*. Section 4.2). Our highly accurate detection and tracking approach allows us to provide additional knowledge to the user: how long is the shoreline, where and when does it end, does it continue further afar after a gap as well as which direction and distance will it continue in. In combination with shoreline-based routing this knowledge really improves O&M in general, but especially in unknown terrain. Furthermore, we show that such knowledge improves spatial awareness of the surroundings in a small Wizard of Oz experiment.

Figure 2.1: Zebra crossings as used in the city of Pompeii more than 2000 years ago. Raised blocks prevented crossing pedestrians from stepping onto the road, often also sewage and drainage system, while space in-between allowed horse-drawn carriages to pass along. 🅭🄍[2]

2.3.3 Pedestrian Crossings

Pedestrian crossings, *i.e.*, locations with specific facilities that allow pedestrians to cross a street, date back to more than 2000 years ago. Originally, these were raised blocks — the spacing in-between allowed horse-drawn carriages to pass — in order to prevent pedestrians from stepping onto the road itself, which was often also used as a city's sewage and drainage system, a common practice in these days (*cf*. Figure 2.1). Today's importance of crossings and the issues these create for people with visual impairments were studied by Matthews et al. (2014).

While there exist many different types of such crossings, we focus our contributions on the two most commonly found variations: *zebra crossings*, *i.e.*, a type of crossing that has no (pedestrian) traffic lights but distinctive white markings on the road pavement, as well as *pedestrian traffic lights*, a crossing type where the pavement markings often differ from country to country, but which always exhibit traffic lights for cars and pedestrians to regulate traffic flow and prevent accidents.

[2]Creative Commons licensing symbols, *e.g.*, 🅭, are used throughout this work to indicate their usage, here 🅭🄍 for CC∅ 1.0 https://creativecommons.org/publicdomain/zero/1.0/

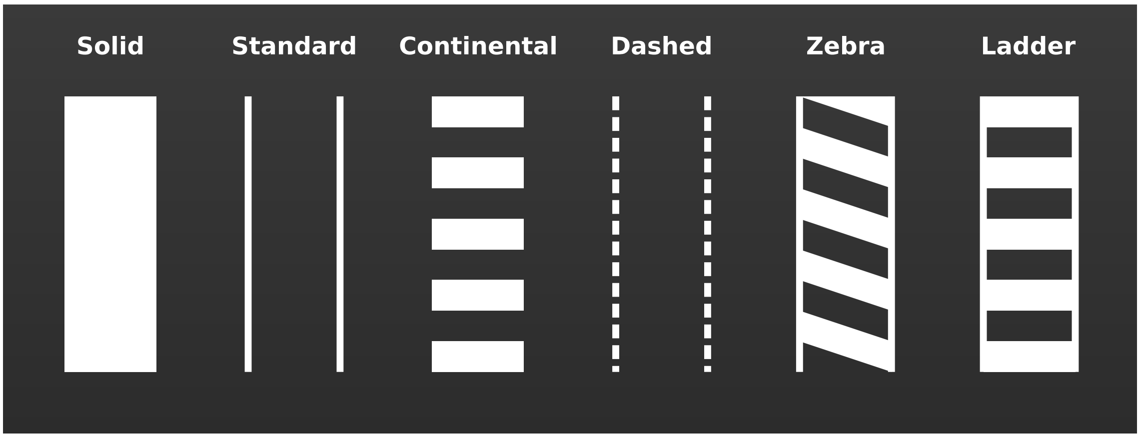


Figure 2.2: Different marking variants for crosswalk pavements as the defined by the United States Federal Highway Administration. Each country uses their own marking style, often contradicting each other, *e.g.*, Germany uses the *Continental* style for zebra crossings.

Zebra Crossings

Zebra crossings are a specific type of pedestrian crossing and are defined slightly different in each country. While in the United States white diagonal patterns including stripes as a border on both sides denote this type of crossing, Germany's zebra crossing resembles their *Continental* style (*cf*. Figure 2.2), *i.e.*, parallel white lines of a certain width. Nevertheless, all zebra crossing types provide no guidance information by themselves and can hardly be recognized using a White Cane, due to them being only flat road pavement markings.

In 1998, Utcke (1998) proposed a grouping algorithm suitable for zebra crossing detection, due to their repetitive appearance. He modeled image feature uncertainties and propagated detection errors to create a robust system that worked well for cluttered real-world images. This approach inspired Se (2000) to improve on it and include it as part of another system, TAPS (Molton et al., 1998a,b). Both approaches were using techniques to group subsequently detected objects, *i.e.*, edges, to recognize zebra crossings from a pedestrian's perspective, but their computationally expensive processing requirements did not allow the creation and testing of any real time prototypes.

Almost at the same time, Shioyama et al. (2002) proposed a system to measure the travel distance of a pedestrian crossing across the road. Their approach was based on the specific width of the crosswalk's ground stripes and their geometric appearance in the image as well as the detection size of a pedestrian light on the other roadside. A few years later their approach was found to be too slow and also required too much manual tuning, so Uddin and Shioyama (2004) proposed an improved version. This very similar system also assumes the specific width of the stripes to be known in advance, *i.e.*, 45cm in Japan, and both approaches only deliver satisfactory results with a very good image quality and marking condition. However, both of these approaches were only focused on the to be expected travel distance of a pedestrian crossing and neither provided any guidance information for the actual crossing, causing Uddin and Shioyama (2005) to suggest another system. This proposed system uses image bipolarity and projective invariants to detect and verify zebra crossings in an image, yielding very good results. Sadly, none of these three developed systems were tested by their authors with visually impaired users in real-world scenarios.

Coughlan and Shen (2006) later used a graphical model, which required much less processing power compared to some of the earlier approaches, to assign stripelets and group individual edges of repeatedly detected patterns. This approach was later improved and integrated into the Crosswatch system, implemented on a smartphone and an user interface was added as well (Ivanchenko et al., 2008b,c).

Ahmetovic et al. (2011) found existing systems were all lacking information with respect to detected zebra crossings' relative location to the user, and proposed the Zebralocalizer system (iOS application), as well as their later improved ZebraRecognizer library (Ahmetovic et al., 2014), the underlying system used for zebra crossing detections. Thus, their approach not just locates zebra crossings in an image taken from the user's phone, but was the first to provide guidance information, as it also communicates the relative location, *i.e.*, distance and direction, with the intent of aiding the user in reaching it safely in the first place.

Shangguan et al. (2014) suggested CrossNavi, a system that detects zebra crossings using computer vision as well, but limits its detection capabilities to actual crossing moments in order to save battery and processing power. However, they found GNSS information to be too imprecise to activate the visual detection, and rely on audio signal processing instead. Problematic is their use of a phone that is mounted to the White Cane, a position that induces blur and limits the cane's mobility, but take pictures only when they assume the cane is not moving at the end of each cane arc.

Their system is able to provide guidance information towards the crossing as well as anti-veering support during the crossing itself.

Lately, Nishikiri et al. (2016) introduced an integrated navigation system that also detects zebra crossings as well as pedestrian crossings. Its novelty is that is was completely implemented on a mobile device, *i.e.*, an Android smartphone, including its computer vision algorithms. Due to the limited processing power and the therefore chosen processing algorithms, their detection accuracy could be considered of moderate performance, however, in the future, with ever improving hardware, it should be possible to achieve much better accuracies.

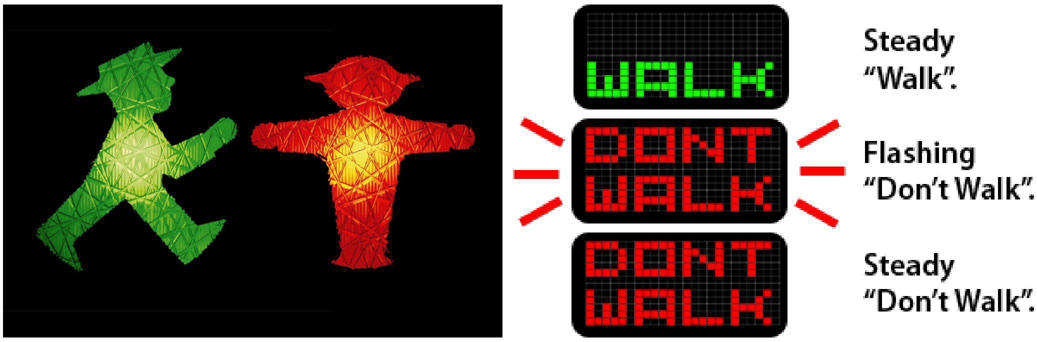
While many systems to detect zebra crossings from a street level perspective have been proposed already, these usually fail at least one or more of the guidelines laid out in section 2.1, *e.g.*, they do not provide relevant location information to the user or do not work in real time. Finally, zebra crossings have only very recently become an area of interest in remote sensing applications, as their exhibited features can also be detected in aerial imagery using the same, or at least very similar, algorithms. An overview of existing approaches for this similar task is provided in section 2.2.1.

Pedestrian Traffic Lights

As shown in figure 2.2, crosswalks can exhibit different pavement marking variants, even within one country alone. Thus computer vision approaches often ignore them and try to detect pedestrian traffic lights directly or using other available information, but these also often differ between countries (*cf*. Figure 2.3). Lately, the detection of traffic lights and their state (red/green) has seen a surge in interest, but mostly for autonomous driving and driver assistance systems, *e.g.*, Weber et al. (2016).

Aranda and Mares (2004) use a color histogram and simple appearance model-based approach to detect pedestrian traffic lights and report on their state and relative position to the camera using acoustic signals. While their system shows promising results, they sadly neither conduct a user study nor report on the quantitative detection qualities of their algorithmic system parts.

Le et al. (2012, 2014) detect a crosswalk's boundary by searching for the white lines that surround it, using normalized cross-correlation template matching. This is sufficiently stable with respect to illumination changes and the authors detect



(a) [3] (b) [4] (Brett Caven)

Figure 2.3: (a) Pedestrian traffic light symbols often seen in Germany, first used in the German Democratic Republic. (b) Common walk light signal found in the United States.

patches that exhibit a near vertical or diagonal black and white border. In a next step, candidates are grouped along straight lines using RANSAC, limited to a sufficient amount of detections along these lines and the detected crossing section is returned. In a follow-up work, they add a vanishing point detection and apply similar approaches to lane boundary detection in unstructured environments, *e.g.*, walkways that contain a sufficiently different textured border. However, none of these approaches were able to run in real time on mobile prototypes, so no user studies were conducted.

Coughlan and Shen (2012) planned to extend their Crosswatch system used for zebra crossing detection (*cf*. Section 2.3.3) to also provide assistance to people with visual impairments on road intersections. To this end, Murali and Coughlan (2013) proposed an approach that requires a user to take a 360 degree panoramic image from their current location. Using this panoramic image, they calculate an inverse projection to acquire a composite image, similar to aerial imagery of that intersection, and match it to a template generated from real aerial imagery obtained via Google Maps. Their overall location error is much less compared to pure GPS-based positioning. Furthermore they gain the ability to estimate the direction of traffic lights and the walking direction across the two adjacent roads and in combination with the phone's built-in IMU can orientate the user accordingly (Fusco et al., 2014a,b). However, such functionality was, to the best of our knowledge, never integrated into Crosswatch.

[3] https://de.wikipedia.org/wiki/Ampelmännchen
CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0/

[4] https://en.wikipedia.org/wiki/Pedestrian_crossing
CC BY 4.0 https://creativecommons.org/licenses/by/4.0

Mascetti et al. (2016a,b) also integrated a pedestrian traffic light detection into a cell phone. In order to deal with different illumination during day and night time, they fix the exposure settings using a small set of pre-determined values supposed to optimize the colorful light emitted by pedestrian traffic lights. As a second step they try to match bundled regions of specific colors to templates of light emitting figure contours and prune not matching candidates. Finally, they use the contour's size and position in the image to remove impossible locations, *e.g.*, under the horizon, and notify the user about the detected traffic light's distance, direction and state.

Roters et al. (2011) developed the first real time capable traffic light detection systems in real world conditions on a Nokia N95, in 2011 a phone used by many people with visual impairments due to the available accessible applications. They search for red and green spots in the image using color filters as well as shape and classify their specific state, both refined and made as easy as possible to compute, so that their algorithms could run on the phone in almost real time.

At the same time Ivanchenko et al. (2010) also proposed their system, developed on a Nokia N95 as well. They search for traffic lights that are close above the horizon, requiring the phone's camera to be held near a horizontal angle, a process they assist the user in, and identify the pedestrian traffic light's state at close to one frame per second.

Very recently Cheng et al. (2018) proposed to train an SVM on Red-Green-Blue-Depth (RGB-D) data to detect crosswalks, traffic lights and pedestrians. A first system test with a single subject shows promising results at six different intersections.

Our Contribution. All existing works in this section focus on a single crossing aspect and for the most part on its detection only. To the best of our knowledge, we created the first system that combines these two very distinct crossing aspects — zebra crossings and crossings with pedestrian traffic lights — in a real time capable system (*cf*. Section 4.3). Furthermore, we not only detect these, but also lead up to and actively assist during their crossing. Finally, our approach is the first one integrated into a navigation and routing system. This is especially important in order to provide a blind user with exact, complete and sufficiently detailed instructions on how to get from one point to another, providing improved GNSS navigation information about turns and crossings and additionally fine-grained information required during pedestrian crossings as well as before and after the crossing itself.

2.4 Influential User Studies

In order to thoroughly test the usefulness of a proposed assistive system, after the required quantitative numerical evaluation of its individual parts has been performed, it often becomes necessary to conduct a qualitative analysis as well. Not all, but many of the discussed works have conducted such user studies to prove their contribution to specific Orientation & Mobility issues. Here we present a selected overview of especially those user studies — further information on these can be found in this chapter's other sections — that have influenced us in the selection of assistive situations to provide assistance for, as well as our overall design decisions and included integration details. In section 2.5, we list key concepts and our most important takeaways that guided us throughout this thesis and influenced all of our designs and implementations.

The generation of custom directions for people with cognitive impairments was analyzed by Liu et al. (2009). Although the authors did not focus on people with visual impairments, their study suggests that navigation is a highly personal experience, not just due to personal impairments, but also due to personal mobility and risk averseness, all highly personal capabilities. They suggest a few considerations when designing O&M systems, where the most important ones to us are: consider individual preferences w.r.t. direction types, leverage a user's familiarity with the surroundings, provide sufficient detail in a single direction, consider personal abilities and conditions, detect errors made and try to intervene, as well as always customize for safety first.

Personal abilities were also found to be a key factor when designing assistive devices for people with visual impairments by Manduchi et al. (2010). While a camera's field of view was identified as a major design choice for computer vision-based assistive systems, it was also noticed that personal ability should be considered in all design stages and technologies used as well. In a follow-up user study, Manduchi (2012) further investigate these issues and report that while computer vision was found to be a viable tool for assistive systems, much care has to be taken w.r.t. to its specific design and implementation choices, *i.e.*, where exactly to place the camera, how to ensure proper alignment, what field of view to choose for a specific application and where to place visual targets in the first place.

Guy and Truong (2012) noted in their user study that people especially wished for assistance in unknown areas or when unexpected events caused them to leave their memorized route, but also note the positive effects that a rich set of information can make in these situations. Moreover, they noticed the internal creation of a perceived mental map by the participants and that some even felt more included into the sighted community. Most interesting was also their usage of cardinal directions, as the city's grid where the study was performed, *i.e.*, Toronto, is laid out at an angle and "north" is really "north-east", a fact which confused people with local knowledge.

Around the same time Zeng et al. (2012) studied a different mobility approach, *i.e.*, a TOF camera and a tactile display. They focused on communicating a spatial layout, so that routing decisions could be made as early as possible, and noticed that while using their systems increased the time taken when training was done in a short amount of time, it also considerably improves safety. Furthermore, they discuss the limitations of TOF systems, as some materials absorb near infrared and some objects might be too small or fast to be detected.

Fiannaca et al. (2014) analyzed how to improve the traversal of large open spaces and prevent veering off. They specifically compared sonification to text-to-speech output and found that for veering both worked similarly efficient, possibly due to the fact that a sonification had to be learned, while text-to-speech seems more natural in the beginning. Moreover, study participants noted the lack of obstacle detection and that (for now) only doors could be used as a reference point.

The fact that the White Cane will be almost impossible to replace was, similar to many others working in this domain, stated by Bhowmick et al. (2014), when they analyzed their IntelliNavi system. They, too, decide on complementing the White Cane, instead of trying to replace it, as many other research prototypes have aspired to, without long-term success. Additionally, they noticed usability issues when a prototype is not fully real time capable.

A different approach to O&M was proposed by Guerreiro et al. (2017), who used route pre-planning as a first step for navigation issues. They concluded that such pre-planning — they refer to it as *virtual navigation* — aids in the creation of an *a priori* mental representation that sequentially lists navigation steps, and is especially helpful when making errors and having to restart from a different point. They also suggested that user perception of distances had improved using virtual navigation.

The prominent NavCog3 system was also examined in great detail by Sato et al. (2017). They reported many factors that are important for a successful O&M assistive system: sufficient accuracy, where *sufficient* strongly depends on the chosen objective, spatial awareness, as in providing personally relevant landmarks and POIs, augmentation of and integration into existing navigation skills, so "users can keep their routine," to support large-scale environments to be widely useful, and familiarity with a system.

Very recently, Avila Soto and Funk (2018) further analyzed public acceptability of their proposed drone-based guidance system. They found that while the general public often accepts such technologies, *i.e.*, if they clearly understand its assistive value, they also raise some issues with respect to passenger safety and the overall perception of people with visual impairments, a major acceptability issue for these themselves, as they would probably not take it around all the time.

Furthermore, the insights provided in section 2.1 as well as many discussions with other researchers and especially Orientation&Mobility trainers[5] have also influenced our own decisions and implementations, as well as provided us with some initial motivations and the necessary awareness in order to create most of our assistive systems in the first place.

2.5 Key Concepts & Takeaways

In this final section we describe our identified key concepts and important takeaways that we relied on and tried to adhere to when developing our own approaches. These are kept short and to the point on purpose, also their ordering is largely irrelevant[6] as we do not value one over the others, they are all considered equally important. The reasoning for most of these can be gained by looking at the preceding sections of this chapter, but especially section 2.1 and section 2.4. Of course, we could not always follow through on all of them in our own proposed assistive systems and their implementations, but had to eventually sacrifice some in order to fulfill others, *i.e.*, our user study prototype required a mobile laptop with a sufficiently powerful GPU to work in real time, which is neither easy to carry around, limits mobility, not

[5] Many thanks to Christoph Erbach and Ottmar Kappen from the "Sehwerk" team for all their shared insights. https://www.sehwerk.com

[6] Just note that the first letters, in the provided order, conveniently create the very catchy acronym DIIPLOMAATS.

readily available everywhere, nor unobtrusive. Please also note that although these concepts are mostly orthogonal to each other, some cannot be clearly separated, as they are heavily intertwined or interdependent, and sometimes even contradict each other as well, *e.g.*, *Prototypes* are limited by existing *Technologies*, which in turn might not be cheap enough for general *Availability*, not unobtrusive enough for wide *Acceptance*, or even create *Safety* issues due to unrealistic user expectations.

Demand Start to identify actual demands first to provide relevant assistance. Do not find solutions for otherwise easily solvable problems, *e.g.*, changes in legislation.

Individual Consider individual abilities. Present options for customization. Provide sufficient details, *i.e.*, only as many details as necessary.

Interface Prefer simple intuitive interfaces. Do not confuse the user. Possibly reuse familiar interfaces. Prevent mental exhaustion.

Prototypes Do not limit a user's mobility and work in real time. Consider wearability to keep the hands free. Use wireless technologies. Apply the *KISS*[7] principle.

Location Improve spatial awareness whenever possible. Present only reliable information. Combine high-level and low-level information.

Orientation & Mobility Deeply integrate O&M techniques. Improve guidance and orientation, never contradict any O&M methods, *i.e.*, consider the White Cane.

Availability Use off-the-shelf components. Use economically available components. Allow as many people as possible to benefit from developed assistive systems.

Acceptance Prefer unobtrusive technologies. Broad acceptance in the community is required to someday become a useful tool.

Technologies Carefully consider technologies with respect to reliability, working conditions, weight, battery lifetime, unobtrusiveness, and other functionalities.

Safety Use safe defaults. Clearly communicate accuracy and limitations. Create reliable and versatile systems. Deal with individual errors, *e.g.*, veering.

[7] "keep it simple, stupid", a design principle coined by the United States Navy in the 1960s.

Chapter 3

Orientation

> "I may not have gone where I intended to go,
> but I think I have ended up where I needed to be."
>
> – Douglas Adams

Orientation&Mobility are always considered in tandem by O&M trainers when they educate people with visual impairments about its techniques (*cf*. Chapter 4). Ultimately, while the mobility aspect allows to "roam around", the orientation aspect of O&M teaches spatial awareness about the current place, *i.e.*, "Where am I?" and "Where do I want to go?", as well as "What should I be aware of?" General awareness of one's own position within the environment helps to understand relationships between the own body, the surrounding space, and the time it takes to cross it. Spatial awareness provides such important cues about the place one is currently traversing, while knowledge about the position and a sought out destination is equally important for an assistive system in order to provide useful information.

However, even latest generation navigation systems only provide high-level information, *e.g.*: "Follow this road," "Turn left in 100 meters," or eventually "You have reached your destination!" Requirements of people with visual impairments are mostly ignored, as these systems do not provide more detailed information about relevant aspects, *e.g.*, large pedestrian crossings of multiple lanes or strange intersection layouts, and do not consider personal preferences, *i.e.*, to prefer alternative pedestrian crossings (Matthews et al., 2014). Finally, commercially available systems

Figure 3.1: Bronze sculptures allow people with visual impairments to “finger their town for the first time.”[1]This historical city center map for the city of Stralsund is only one of many such sculptures now found in European cities. It displays houses, churches, markets, parks, ponds, the city’s harbour, and other places, to be touched and experienced in an entirely new dimension for sighted as well as people with visual impairments alike. It also exhibits braille encoded information about street names, churches, important sights, and many other places and provides a better understanding of distances and spatial relations.

exhibit a very challenging user interface, as accessibility is rarely considered, forcing people with visual impairments to rely on other alternatives instead (*e.g.* Figure 3.1).

This chapter’s focus is on orientation assistance, *i.e.*, we closely look at some high-level information that is relevant to impairment aware navigation and routing in section 3.1. We further investigate how to integrate such high-level navigation information with guidance aspects. By enriching the routing process with relevant low-level aspects in section 3.2, we decrease the identified existing gap (*cf.* Section 1.3).

Our Contributions. To improve the availability of relevant geospatial data, we analyze aerial imagery and localize zebra crossings (Koester et al., 2016) in section 3.1.2. We then integrate this data into a novel, truly impairment aware, routing algorithm (Koester et al., 2017) in section 3.2.2, capable of combining high-level routing information with low-level guidance aspects, *i.e.*, it integrates available shorelines into the routing process and prefers to use the safest pedestrian crossing available.

[1] https://www.blinden-stadtmodelle.de/en

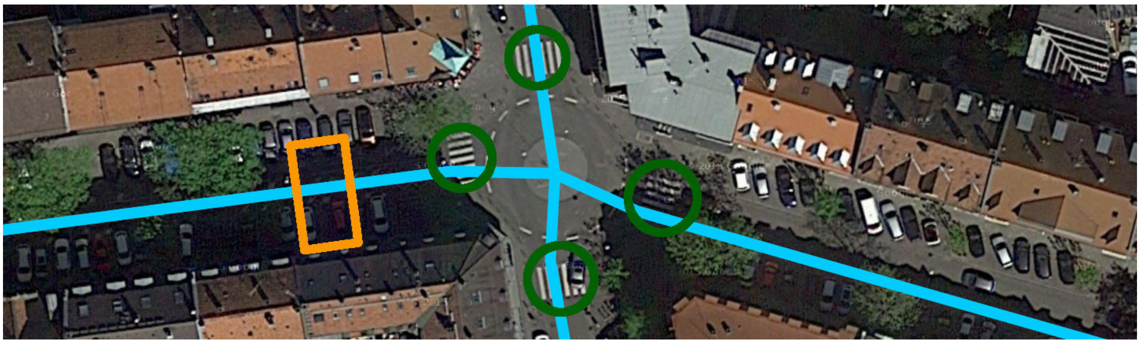


Figure 3.2: Aerial imagery and zebra crossing labels for a roundabout in Karlsruhe, Germany. The road's center lines (bright blue) from OSM metadata often do not perfectly match the imagery, while cars, trees or shadows might partially occlude zebra crossings (dark green circles). Also visible is an example detection window (bright orange rectangle) that searches along center lines for zebra crossing candidates. Aerial imagery © Google Maps.

3.1 Aerial Imagery

In the last decade, aerial imagery has become widely available for the general public to investigate online. Efforts such as "Google Maps," which started its public service in 2005, have been providing ever better resolutions and coverage on top of routing capabilities for road-based driving. Only a few years later, a vibrant community had developed, in order to use this newly found availability — before Google Maps, satellite data and aerial imagery was mostly kept private within mapping companies — in order to create geospatial databases for research and public purposes (*cf*. Section 2.2.2). Computer vision can be a great benevolent factor in such efforts, as many ground level features are clearly visible in such aerial imagery, given a sufficient resolution. Figure 3.2 shows an example of such combined aerial imagery with labeled metadata for road center lines and zebra crossings.

Manual creation of such databases is quite cumbersome and error prone, as volunteers have to look at aerial imagery and label object instances by hand, which often happens only in their own neighborhood, thus data quality greatly varies (Haklay and Weber, 2008). To guarantee a certain quality, distribution and precision requires lots of organisation and manual verification from involved communities. Using semiautomatic, *e.g.*, computer vision-based, approaches to improve the precision and detect relevant objects in the first place, has thus been a long standing research topic, *e.g.*, Quam (1978). This has lead to an improved availability of geospatial databases, which starts at small municipalities and usually ends at a nation wide level. However, each country either creates or requires its own databases, which are often not publicly available, uses non-uniform data storage formats, which prevent an easy exchange, and might even be out of date. Thus, efforts like OpenStreetMap (OSM) provide a great benefit, not just to the general public, as they do not limit themselves to any country's borders and provide unlimited access to *all* of their user generated data.

Yet, very few geospatial information databases exist for the requirements of pedestrians, especially few for those of people with visual impairments, or such databases are missing relevant features. Even where they are available, they are often not integrated into existing O&M solutions and devices. While, for example, pedestrian crossings are only rarely equipped with accessibility aspects, having such information readily available for routing systems could lead to an automatic generation of safer routes. We believe that an increased availability and visibility will over time lead to better routing algorithms and improved integration into created assistive systems, as well as improve routing for pedestrians in general. Maybe such efforts can even raise awareness with city planners, to become aware of and include more specialized requirements into their designs. Such knowledge will also become indispensable for autonomous driving efforts, as it provides great safety benefits for pedestrians.

3.1.1 Pedestrian Crossings

There exist many different ways of crossing a busy street that have become an essential aspect of today's traffic flow. These exhibit a varying degree of accessibility, ideally built-in by design or additionally added afterwards: marked crossings, *i.e.*, crosswalks with or without Accessible Pedestrian Signals (APS) and also zebra crossings, or unmarked crossings, *i.e.*, just crossing the street in any given place (also referred to as jaywalking). Zebra crossings themselves only provide very basic accessibility, as they exhibit only limited tactile feedback to people with visual impairments. Ideally, they are encompassed by tactile pavements as shown in figure 3.3. While their accessibility level is far from perfect, they are still a much preferred alternative to informal crossings (Matthews et al., 2014). Thus, having reliable information about zebra crossings — and of course other controlled crossing types as well — within one's vicinity allows for safer mobility. An overview of the related work most relevant to our own approach is provided in section 2.2.1.

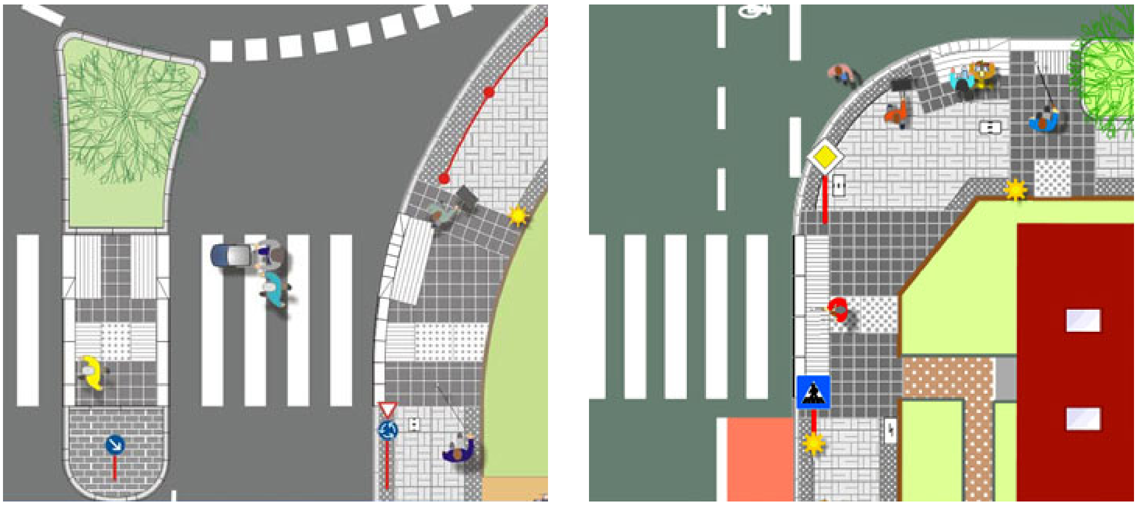


Figure 3.3: Schematic examples of zebra crossings and their encompassing tactile pavements, as part of the pedestrian walkway on both sides of the street. These inform people with visual impairments about the zebra crossing's location (dotted areas) and provide a direction as well (ruled areas), to align before crossing the street. (Dipl. Ing. Wendelin Mühr[2])

3.1.2 Zebra Crossing Detection

Zebra crossing detection in aerial imagery has so far only been proposed by Ahmetovic et al. (2015) (*cf*. Section 2.2.1). We propose to use a purely data driven appearance-based machine learning approach, in comparison to handcrafted features, and to rely on available geospatial data as a weakly annotated data source (Koester et al., 2016).

We rely on available geospatial from the OpenStreetMap (OSM) project. For some areas, coverage is already quite excellent, *e.g.*, the City of Karlsruhe is not only home to the Karlsruhe Institute of Technology, but also to a vibrant OSM community that has not just labeled relevant crossings, but also park benches, trash cans and even individual plants and trees. Based on this weakly labeled data — as it often contains errors or imperfections — we generate a first training corpus of a selected region. We use the OSM "Overpass API"[3] to retrieve a list of already known zebra crossings, using a simple query language (*cf*. Listing 3.1), which is also very similar for roads and their connections, and receive an Extensible Markup Language (XML) file with a list of individual nodes, as well as some meta data along with it (*cf*. Listing 3.2).

In a next step, we use "Google Static Maps" — but this would be possible with any other aerial imagery provider as well — to download the imagery for our received

[2]https://www.barrierefreie-mobilitaet.de
[3]https://overpass-api.de

Listing 3.1: OSM Overpass API example query

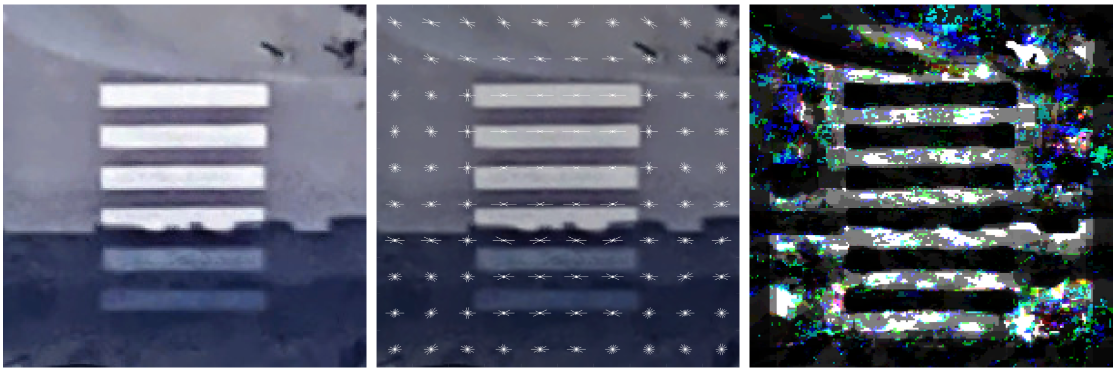
```
[out:xml];
area[name="Karlsruhe"];
node(area)[crossing=uncontrolled];
out;
```

Listing 3.2: OSM Overpass API example result

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="Overpass API...">
...
<node id="14796420" lat="48.9985602" lon="8.3881125">
        <tag k="crossing" v="uncontrolled"/>
        <tag k="crossing_ref" v="zebra"/>
        <tag k="highway" v="crossing"/>
</node>
...
```

zebra crossing candidates' locations. The data could be cleaned for parking areas, factory sites and other uninteresting properties, but this is not required — it just saves downloading bandwidth and computational time. Furthermore, we request roughly the same amount of other crossing types or just general road connection points, usually without zebra crossings, to be used as negative examples, and if their amount is found to be insufficient, we also request linearly interpolated locations, *i.e.*, the road surface between two of these connection points. While an actual zebra crossing can usually be found within the center of the zebra crossing candidate image, the OSM-based meta data's locations are not always perfectly aligned or absolutely precise, so our features should compensate for these slight misalignments and also be able to cope with a few mislabeled instances as well.

It is also beneficial to gain knowledge about the road's direction, in order to automatically align the zebra crossing in this stage and reduce rotational variance, allowing us to require far less data. This can be done using linear geometric calculations that are using the neighboring road's connection points. This in turn allows a sliding window approach to be used later on that explicitly follows the road structure and reduces the required processing time by a large amount, as we do not have to scan all possible locations within a specific region for zebra crossing candidates and can focus on actual road surface instead, again saving on bandwidth and processing time.



(a) (b) (c)

Figure 3.4: Example aerial imagery with (a) a zebra crossing and its feature visualizations: (b) HOG's blockwise computed gradients at a coarse block size for better visibility, and (c) LBPs, where the binary vectors are grouped and visualized by distinct colors, showing the feasibility of these features for this classification task. Aerial imagery © Google Maps.

Features & Classifier

After much consideration, we decided to use Histograms of Oriented Gradients (HOG) (Dalal and Triggs, 2005), *i.e.*, the implementation provided by Felzenszwalb et al. (2010), as well as the Local Binary Pattern (LBP) descriptor (Ojala et al., 2002) as our main features. Especially LBPs can be made computationally inexpensive when using pre-calculated lookups for the different pixel offsets in the image based on its selected parameters, *i.e.*, diameter and neighborhood size. While HOG's processing details, *i.e.*, its block normalization, partly helps to compensate for shadows or general illumination differences, LBPs are also capable to deal with these issues by using relative difference relationships in their neighborhood compared to their center point, and not absolute differences. Furthermore, HOGs describe local intensity gradients and their direction, *i.e.*, implicitly perform an edge detection, which the mostly grey and white, also very regularly striped from an aerial viewpoint, zebra crossing texture exhibits plenty of. LBPs are very suitable for this regular texture as well, as their final feature also consists of a normalized feature vector, but require a sufficiently large size, *i.e.*, a radius larger than ten for this task. Due to its rotational invariance, the LBP's histogram is oriented with respect to its maximum histogram bin, LBPs can also deal with rotational variations should our center line alignment be imprecise. Finally, we investigated a combination of HOG and LBP features, but their mostly negligible improvements came at an increased computational cost. Figure 3.4 shows an example of an aerial imagery zebra crossing and its features.

Using the provided, often slightly inaccurate, zebra crossing location in combination with our interpolated road direction, we can align the zebra crossing to appear within a pre-defined selection window. We then process this window by calculating our features on it, and also use data augmentation techniques, *i.e.*, slight rotations, shifting and mirroring, to increase our training corpus size. All calculated feature vectors are gathered for our entire training corpus and then a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) is trained as a binary classification problem, *i.e.*, the selection window contains a zebra crossing or not. We investigate simple linear kernels, simple dot products which are much faster to train, as well as Radial Basis Function (RBF) kernels — a popular kernel function for various learning algorithms, which is defined as:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad , \tag{3.1}$$

where σ is a choosable parameter that affects the kernel's decision region and $\|\cdot\|^2$ the squared Euclidean distance between training corpus points, which places an increased weight on further apart entries. The RBF kernel is more expensive to calculate, but provides better classification results, as it has a better chance of (almost) clearly separating the individual classes. For now, we just want our SVM to separate these two classes as well as possible, while still allowing for sufficient input variations.

3.1.3 Evaluation

In order to properly evaluate our proposed zebra crossing detection approach, we have prepared a dataset of ∼3100 zebra crossings, collected from various urban and rural regions in Germany, without taking any possible defects into account, *e.g.*, misplaced labels, occlusion, marking deterioration, partly shadowed areas, illumination issues, *etc.*, in order to preserve the variances and individualities that zebra crossings exhibit in aerial imagery. It is based on ortho-imagery, *i.e.*, a format that allows us to calculate per pixel latitude and longitude locations, as it is already rectified in a way such that a single pixel of a tile always covers the same amount of real world space. Referencing latitude and longitude locations is therefore not influenced by the pixel's position in the image, but only by the imagery's zoom level.

We then rely on the same sliding window approach as applied during training. This considers only actual road surface and greatly reduces false positives, as other regular

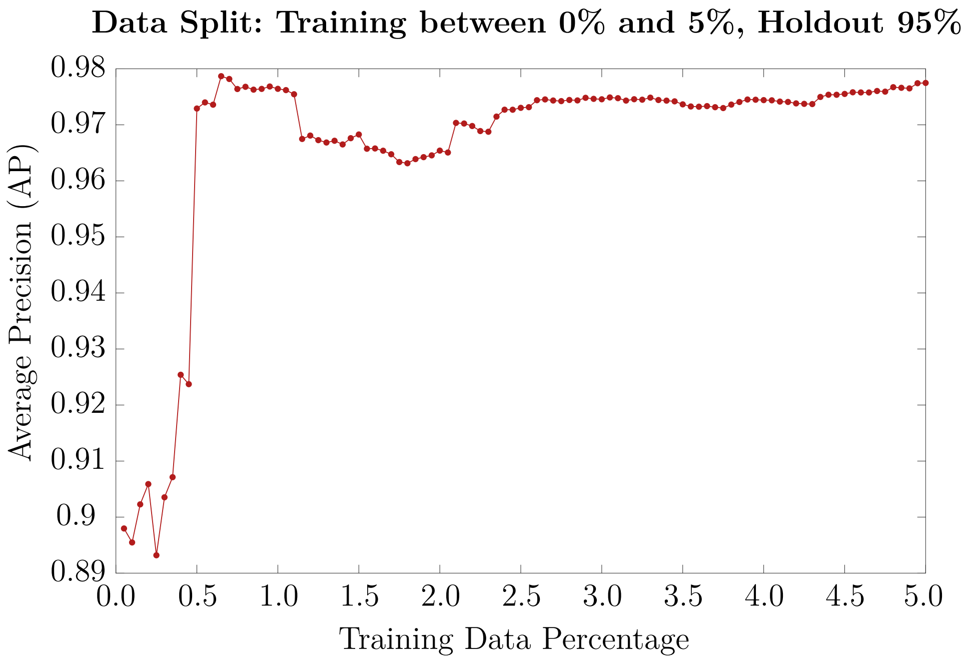


Figure 3.5: Average precision curve for our zebra crossing classification approach when using very little data. We randomly split our dataset (∼3100 positive and ∼3100 negative instances) into a training and a holdout test split. We then train SVMs using between 0% and 5% of all available data, increasing the amount of used data in steps of 0.05%.

structures in aerial imagery with a similar appearance, *e.g.*, rooftops or staircases, are automatically not considered. Furthermore, we can decide to rescan the same region for yet unlabeled zebra crossings, also improving the labels of our own training corpus, or apply the resulting classifier to complete disjunct regions.

In a next step, zebra crossing candidates are clustered using either non-maximum suppression (Canny, 1986) or straightforward linear interpolation between the highest scoring pair of two successive zebra crossing candidates along a road surface center line. The latter clustering approach yields a slightly improved location accuracy, although it requires sufficiently small detection windows and sliding window steps, which significantly increases the amount of selection windows that have to be classified. However, even a step size of half the selection window's width, *i.e.*, a 50% overlap between sliding windows, was already found as sufficiently small for our task.

Required Amount of Training Data. We first analyze how much data our approach really requires for a successful classification (*cf.* Figure 3.5). To this end, we randomly split the data into a training set (5%) and a holdout test set (95%). We then train our classification pipeline between 0 and a maximum of 5% of all the

Table 3.1: Zebra crossing classification results for our own dataset. "{HOG,LBP}-LINEAR" uses a linear, "...-RBF" a RBF kernel. "HOG^{30x30}" refers to the used *block size*, while "LBP$^{17/10}$" refers to *radius/neighbor* variations. All results are listed in %.

Method	Precision	Recall	Accuracy	Avg.-Prec.
HOG^{30x30}-LINEAR	74.8	93.1	92.4	94.43
HOG^{20x20}-RBF	95.2	96.2	**98.9**	97.99
LBP$^{17/10}$-LINEAR	**99.4**	**97.4**	98.4	**99.56**
LBP$^{17/10}$-RBF	**99.7**	97.0	98.3	**99.56**
HOG^{25x25}-LBP$^{17/10}$-RBF	98.1	**98.8**	**99.4**	**99.59**

available data (0% to 100% of the training split), increasing the amount of used data in 0.05% steps. Figure 3.5 shows that even when using very little data, *i.e.*, only 0.05% of our dataset, we already yield an average precision of almost 90%. Moreover, at 0.5% our classifier already performs almost at its best overall average precision. Further increasing the data yields different behaviors, as sometimes this might already include a mislabeled instance in our automatically created dataset, or confuse the classifier by training on rather rarely occurring instances (outliers). However, this process already stabilizes at around 2.5% and adding further data to the training process does not significantly change the average precision anymore.

k-Fold Cross-Validation

As it is far easier to create negative samples than to collect more actual zebra crossings, we further analyze our classifier in a k-fold cross-validation, where we train on all folds except for one and use that for testing. We also use classical data augmentation techniques, *i.e.*, minor rotations, mirroring and subtle translational shifts. Our k-fold cross-validation ($k = 5$) thus contains an almost equal amount of randomly chosen positive and negative samples. We also considered *hard negative mining*, where incorrectly classified samples are included in the training process as well, but our already very high precision does not require it. We test a number of parameter configurations, *e.g.*, HOG block size, LBP neighborhood and radius, as well as different SVM parameters. A first overview of classification results can be seen in table 3.1 and figure 3.6 for different parameter combinations. These results show that our algorithm is capable to achieve an almost perfect performance, considering our weak labels. Also notable is the effect of parameter tuning: While the chosen

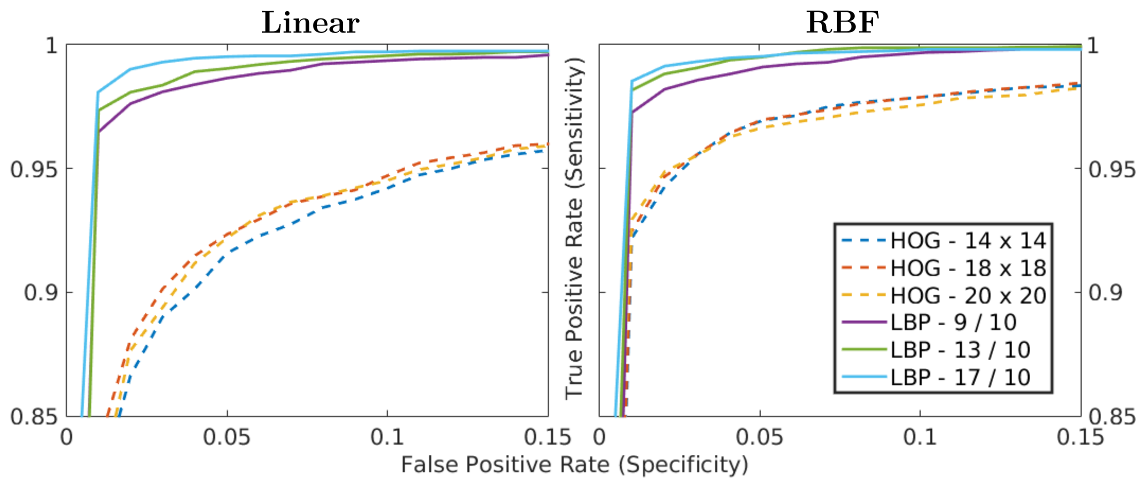


Figure 3.6: Receiver Operating Characteristic (ROC) curves for our zebra crossing detection. We perform a 5-fold cross-validation on our own data, comparing linear kernels, where the HOG feature performs much worse than LBP, and RBF kernels, where the selected feature and its specific parameters make less of a difference.

feature(s), *i.e.*, HOG *vs.* LBP, show a large performance difference, similar to as the SVM kernel type largely influences the performance, the differences between specific parameter combinations for each feature is almost negligible (*cf*. Figure 3.6).

Comparison to State of the Art

The only other proposal for zebra crossing detection in aerial imagery was by Ahmetovic et al. (2015), who suggested their approach just shortly before us. Similar to our proposed algorithm, the authors rely on Google Static Maps aerial imagery, but additionally use Google Street View panoramaa, yet also — like us — download only map tiles that contain road surface. They then use a cascaded classifier with two stages: first extract candidate crossings from aerial imagery, second validate these in nearby street view panoramas (*cf*. Figure 3.7). For their reported numbers, the authors manually fine-tuned their algorithms for maximum recall to minimize the amount of missed zebra crossings, at the cost of a slightly lower precision.

For the first stage, they download map tiles that contain road surface, then they use a line segment detection algorithm, EDLines (Akinlar and Topal, 2011), and group detected segments into sets of stripes based on their horizontal and vertical distance as well as a parallelism criteria, which ensure the detection of parallel groups of segments within an immediate vicinity. They then validate these line segment groups

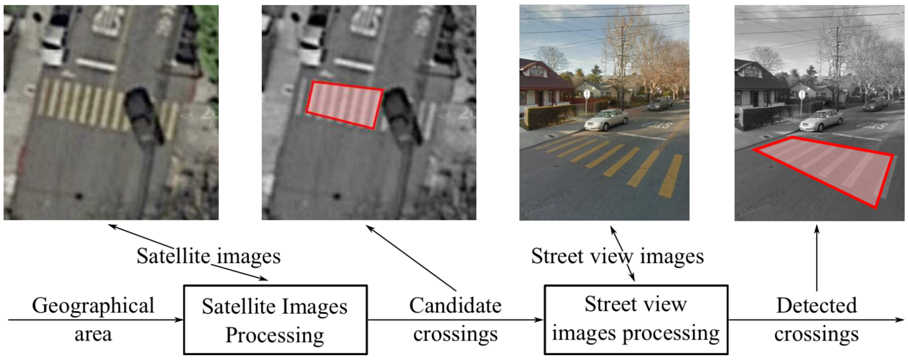


Figure 3.7: Overview of Ahmetovic et al. ZebraSpotter's two-stage approach that relies on aerial imagery as well as street view panoramas and uses a line detection algorithm-based on their ZebraLocalizer library for both steps. Figure taken from Ahmetovic et al. (2015).

based on their number of stripes and a color intensity feature, as such markings usually exhibit a stark color contrast to the normal road pavement. In this stage's final step, they merge these line segment groups into candidate crossings, based on the segment group's coordinates and direction, with existing candidates, *i.e.*, if a nearby candidate was previously found. They adapted this technique from their previous work on ZebraLocalizer (Ahmetovic et al., 2011).

Their second stage verifies these candidates in street view panoramas. For each candidate, they request nearby panoramas within a fixed distance limit to deal with occlusions by cars or other objects, and use a fixed camera pitch and field of view settings to reduce the amount of downloaded panorama pieces as well as to simplify a following image rectification step. They again use ZebraLocalizer to detect crossings in these street view panoramas, until the candidate is verified or all nearby panoramas were processed. Detections in the panoramas are again matched to the candidate's coordinates, but might have some inaccuracies due to GPS imprecisions, so they consider a fixed distance tolerance when comparing coordinates and angular deviations of possible candidates. This second stage either validates a zebra crossing candidate, when a panorama contains a close enough match, or, after all nearby panoramas were processed, discards this candidate and proceeds with the next one.

Comparison. In order to present a fair comparison between our approaches, we have downloaded the exact same San Francisco neighborhood from Google Maps using our already discussed approach, limited to actual road surface. However, the individual grid cells of the city pattern where often so tiny that a merging of all acquired tiles would lead to an almost complete aerial image as well. We also manually searched for

Table 3.2: Zebra crossing detection across countries results, where Ahmetovic et al. provide numbers for 141 zebra crossings they found in a San Francisco area. We compare ourselves to both of their stages, *i.e.*, aerial imagery (AerImg) and street-view (StView) based detections, as well as their combined results, once based on surrounding data (Ours $_{SanFrancisco}$) and on our own dataset only (Ours $_{Germany}$). All results are provided in percentages and optimized for highest recall, as Ahmetovic et al. did.

	AerImg	StView	Precision	Recall
Ahmetovic et al.	✓	✗	68.8	**97.2**
Ahmetovic et al.	✗	✓	**97.2**	**97.8**
Ahmetovic et al.	✓	✓	**97.2**	95.0
Ours $_{SanFrancisco}$	✓	✗	96.2	95.7
Ours $_{Germany}$	✓	✗	**98.9**	38.4

∼250 zebra crossings within a disjunct San Francisco neighborhood as well as some other cities in the San Francisco Bay Area to prevent an accidental data overlap. OSM zebra crossing metadata sadly was not available for the San Francisco area, which is exactly one of the main motivations for our approach. Also, the authors did not publish their used ground truth, but instead included only a figure. Thus, we had to manually label the selected San Francisco rectangle's zebra crossings: We counted 184 zebra crossings overall, where 122 are *continental crossings*, *i.e.*, a typical crossing type found in the United States very similar in appearance to the German zebra crossing, as well as 62 crossings that also contain *transverse markings*, *i.e.*, two white lines oriented perpendicular to the stripes, which create a ladder-like appearance (*cf*. Ahmetovic et al. (2015)). The numbers and locations we base our evaluation on differ slightly from Ahmetovic et al., due to us considering ladder style crossings also as zebra crossings. Furthermore, we also ignore the 110 plain *transverse crossings* as Ahmetovic et al. did (*cf*. Figure 2.2 "Standard" variant), which are found on many other street intersections and only consist of two parallel lines that mark the region to be used by pedestrians while crossing the street.

We compare our two approaches in table 3.2, *i.e.*, an across countries pre-trained classifier based on German data alone, as well as a classifier trained only on nearby San Francisco data, that had to be collected manually. It must be noted that we can only compare to their published results, *i.e.*, their recall optimized results for both stages of their detection pipeline. However, for aerial imagery alone, we achieve a much better performance (Ours $_{SanFrancisco}$), *i.e.*, a precision of 96% *vs.* their 69%,

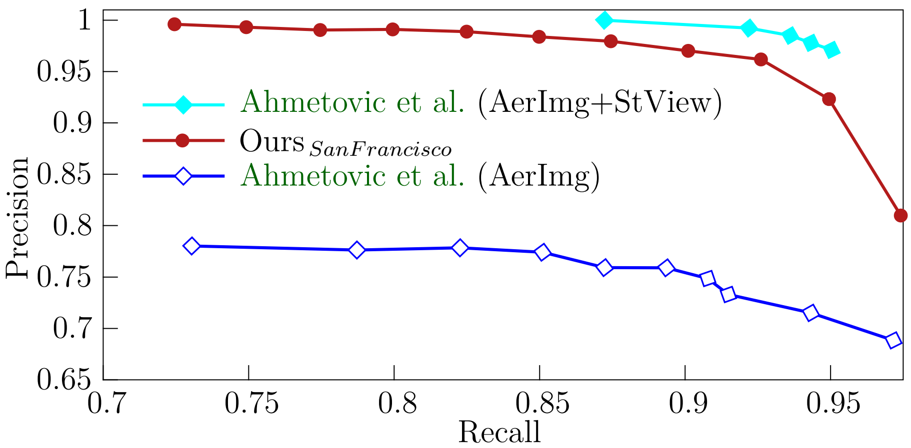


Figure 3.8: Our own classification result for the same San Francisco rectangle as compared to Ahmetovic et al.'s performance. We used their original figure, included our own (discretized) data and slightly modified it for an improved readability.

while exhibiting almost the same recall. Furthermore, our across countries classifier trained on German aerial imagery alone (Ours $_{Germany}$) already performs reasonably well, yielding the best precision of all compared approaches at almost 99%. Even at a comparatively low recall of only 38%, it supports our intention that this classifier would be capable to assist in the creation of a large training corpus for an unseen region, with a high accuracy and thus only a few weak labels, which could even be reduced in numbers by using only the top scoring detections for further training.

Figure 3.8 presents an additional comparison of both approaches. We further compare our own purely data-driven classifier, trained only on ∼250 nearby zebra crossings from the San Francisco Bay Area, to their two detection stages. We show that our aerial imagery-based approach (red line) already achieves much better results compared to their first detection stage (dark blue line). Additionally, we almost perform as well as after their second stage (bright blue line), where they detect candidates in aerial imagery and additionally validate these in street view panoramas. We therefore require much less data for a very similar final performance, especially without the use of street view panoramas, which are also often not available for all regions. Furthermore, the combination with our across countries classifier allows us to first search for zebra crossings in an unseen region, fully automatically and with a very high precision, and then use that data to train an improved classifier in a second step, creating a final classifier that almost beats their hand engineered one for this specific region fine-tuned approach. A classifier built out of both these steps would prevent much labour intensive manual search and labeling work.

3.1.4 Conclusion

We have proposed an appearance-based machine learning approach for zebra crossing detection in aerial imagery. We compare ourself to the only other available related work (Ahmetovic et al., 2015), a line-based search approach, and achieve state of the art results on aerial imagery alone compared to their two-stage algorithm. Our algorithms are widely usable and automatically learn the region specific appearance of a zebra crossing. This is especially important, as different regulations for zebra crossings in different countries (*cf*. Figure 2.2) would require the manual creation of large datasets as a training corpus for automatic systems. However, our algorithm requires comparatively little data to begin with, already delivering a very promising accuracy, which is why we do not use a deep learning approach for this task, as requiring more data defeats one of our main objectives.

Additionally, our algorithm is capable of detecting yet uncharted zebra crossings and further allows us to integrate these into existing databases, which can then finally be used for specialized routing algorithms. It also enables the creation of classifiers to be used across countries, in order to examine completely new regions, where no efforts so far have been undertaken. Therefore, it allows for a process that could significantly speed up such attempts by providing a base training corpus, where errors could easily be corrected by a human or an automatic system — an approach that is much faster to implement than to have to start searching for zebra crossings in aerial imagery from scratch. We believe that an improved availability and accuracy of such geospatial databases will strengthen their usage in general routing applications in the long term, especially in guidance systems for people with visual impairments, and thus also improve overall pedestrian safety.

Finally, we also investigated further relevant ground level features, *i.e.* tactile pavings, but the resolution was found insufficient and it was impossible to retrieve any geospatial data about their precise locations for weak labels, although they are starting to become included in OSM data as a metadata key for pedestrian walkways, and efforts[4] are underway to include them as a precisely located object as well.

[4] https://wiki.openstreetmap.org/wiki/OSM_for_the_blind

3.2 Routing

The most common algorithm to identify the shortest path in a graph network is the "Dijkstra-Algorithm", named after its inventor Edsger Wybe Dijkstra (Dijkstra, 1959). Even today, it is still used in various applications, *e.g.*, packet switching to find fastest connections between telecommunication nodes, or less prominently, the shortest path to a given destination for traffic navigation. Although such routing systems have been available to car drivers for decades now, they only became available to pedestrians within the last ten years, especially with the growing ubiquity of smartphones, *i.e.*, pedestrian routing was first integrated into the Google Maps application in 2015.

Outdoor as well as indoor navigation creates great challenges for people with visual impairments. Urban surroundings are getting ever more complex in order to accommodate the many different forms of mobility. Meanwhile, urban element design only rarely accounts for specific needs, which might also contradict each other, *e.g.*, having a lowered curbline at an intersection to allow people with physical disabilities to cross the street contradicts the requirements for people with visual impairments, who might use a raised curbline as an outer shoreline, and additionally to position themselves for the crossing. It is common for pedestrian walkways to not exhibit any tactile pavings, except for very few locations, *e.g.*, around zebra crossings or on tram stations. Furthermore, urban elements such as trash cans, lamp poles or safety posts are usually placed where they provide the greatest convenience and benefit for the general public. However, this might contradict Orientation&Mobility techniques as well, especially when they are placed close to shorelines or at crossings. Such obstacles often impede the process of mobility for people with visual impairments and furthermore provide a major source of confusion as well as safety issues.

Pedestrian crossings also only rarely exhibit advanced accessibility features, *i.e.*, pilot tones, acoustic and haptic walk light phase notifications, or relief symbols that precisely describe what to expect and look out for during the crossing (*cf.* APS). Furthermore, just navigating the general street layout, *e.g.*, round-abouts, or their even rarer counterparts, through-abouts, construction sites, or busy places, already presents a very demanding challenge. While GNSSs have greatly improved the overall confidence in urban navigation settings, the increase in mobility is at the same time limited to very high level information only — they do not provide obstacle information and guidance on a fine level suitable for people with visual impairments.

New GNSS technology, *i.e.*, "Galileo,"[5] promises an average accuracy of less than a meter, compared to ten meters for current systems. Such a precision would allow for more fine-grained applications that could rely on very precise positioning, even in narrow city street canyons. Nonetheless, mostly POI-based, commercially available assistance systems have been created already, for example those by "Humanware,"[6] the hugely popular "Trekker Breeze," more recent "Victor Reader" series devices, or "BrailleNote GPS" devices and applications, which can navigate along pre-recorded routes. Also, "BlindSquare"[7] has become a highly popular application within the community, as it reads out POIs, pedestrian crossing information including street names, as well as personal favourite locations. It provides a specialized interface to "Foursquare"[8] and also relies on its database, an ever growing corpus of crowd sourced information about businesses, POIs, sights and other user supplied content, thus improvements made there also benefit people with visual impairments as well.

3.2.1 Impairment Aware Routing

What people with visual impairments — as well as people with other forms of disabilities — really require is a routing process that is better suited towards their specific needs, which integrates these needs into its decision process. For people with visual impairments, one possible adaptation could be routing on a shoreline level of detail, modeled after Orientation&Mobility (O&M), *i.e.*, White Cane techniques.

Such an adapted routing could provide measurable safety benefits, as it relies on less unmarked crossings and favors more accessible alternatives instead. This improves the user's safety, the individually perceived as well as the actual physical one (Matthews et al., 2014). Furthermore, upcoming route issues can be communicated early on and might cause the user to prefer a completely different route altogether, preventing dangerous situations before they even arise. Finally, with such an improved routing people with visual impairments can acquire a better understanding of their immediate environment, as it increases their spatial awareness of their surroundings.

Schmitz and Ertl (2014) have already proposed map-based transformations to generate a pedestrian walkway network out of roads and allow routing along *hypothetical*

[5] https://galileognss.eu
[6] https://www.humanware.com
[7] https://www.blindsquare.com
[8] https://foursquare.com

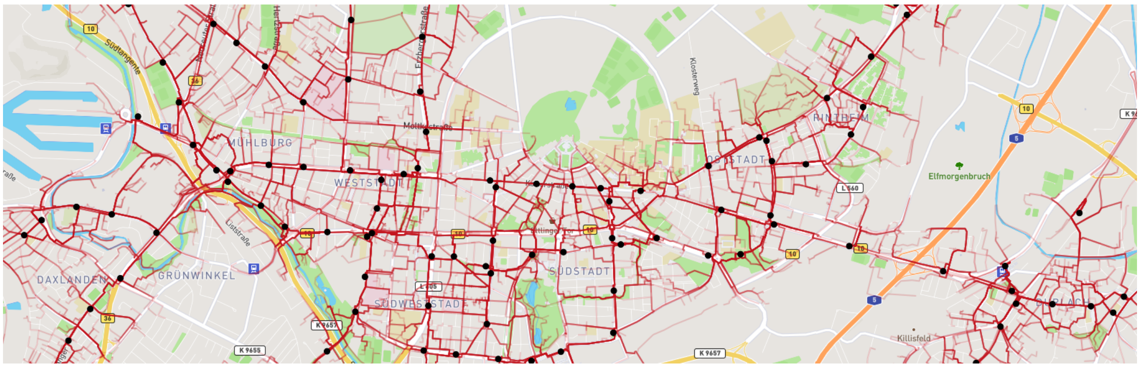


Figure 3.9: A routing accessibility heatmap for the City of Karlsruhe in Germany, with semi-randomly generated routes: Public transit stations (black dots) are used as start points with targets within their immediate neighborhoods. This represents a common mobility scenario, not just for people with visual impairments. Stronger colored segments imply that these were integrated into our generated routes more frequently, signaling that they are considered more accessible than alternatives. Map data © OpenStreetMap contributors.

pedestrian walkway locations, and a further in-depth discussion of related work relevant for our approach is presented in section 2.2.2. Inspired by our prior work on aerial imagery zebra crossing detection (*cf*. Section 3.1.2, Koester et al. (2016)), we instead propose to use available geospatial databases of important O&M components, *e.g.*, building facades and others, to be integrated into the routing process (Koester et al., 2017). To this end, we utilize publicly available data sources, *i.e.*, OSM, to gather information about roads, building facades, or pedestrian crossing locations. We then present a customized routing algorithm w.r.t. our identified restrictions and main criteria, developed in accordance with O&M trainers:

- Avoid informal crossings as long as possible;
- Prefer Accessible Pedestrian Signals (APS) over inaccessible pedestrian signals;
- Follow along shorelines whenever they are available;
- Consider distance trade-off for an increase in perceived and physical safety.

Our approach generates safer routes, while a minor increase in distance is negligible compared to the perceived and physical safety gains. A heatmap showing preferable routes in an urban area, generated with our novel routing approach, is presented in figure 3.9. We first evaluate different routing algorithm prioritizations that focus on different objectives, as well as a combined version. Furthermore, we perform a small *Wizard of Oz* experiment (Dahlbäck et al., 1993; Kelley, 1983, 1984), *i.e.*, a blind person follows along pre-generated routing instructions along two unknown routes.

3.2.2 Generating Specialized Routes

We propose to generate fine-grained routes that integrate the O&M shorelining technique and prefer accessible controlled crossings, based on available OSM meta data about relevant crossings. OSM provides such data, albeit with a varying degree of completeness and precision (Haklay and Weber, 2008), due to its collaborative nature. However, its quality has steadily improved since its creation in 2004.

A project that uses OSM data for routing purposes is the Open Source Routing Machine (OSRM). It provides a public user configurable routing engine, used for example by a mobile application created by Luxen and Vetter (2011). In our experiments, we exclusively rely on OSM data and the OSRM system, but our approach of course also transfers to other data sources and other routing systems. Especially having highly reliable data available, such as the data used by land registry offices around the country, would allow for even further improvements to be made.

First, the OSRM engine calculates a road network-based route that is very coarse, *i.e.*, consists mostly of road intersections. However, this process defines a search window on the map for our next stages and reduces computational complexity. While we consider urban areas exclusively, OSRM routes still often contain deficiencies, *e.g.*, road-based routing, a change of roadsides is not considered at all, very large intersections are easily crossed within a single routing step, and many others that cause such routes to become inappropriate for pedestrian navigation. In a next step, additional details are considered by our specialized routing algorithm. This two-stage approach significantly speeds up route searching, especially for larger distances, as it first looks at the route from a high level perspective and then refines it into smaller segments suitable for navigation of people with visual impairments.

Directed & Edge Expanded Graphs. Meta data acquired from OSM usually consists of a list of (XML) nodes, *i.e.*, points with longitude, latitude and other information, such as connections to other nodes or object type specifications (*cf.* Listing 3.2). Such nodes and their connections can be easily transformed into *Directed Graphs* — a graph with directed edges that usually have weights or costs associated to them — and stored as large matrices, a much more suitable presentation and storage format for routing algorithms to work on. While this models the connections between points, its edges can only contain information about the edge itself, *i.e.*, in our case the type of road, its distance or its speed limit. However, it does not store

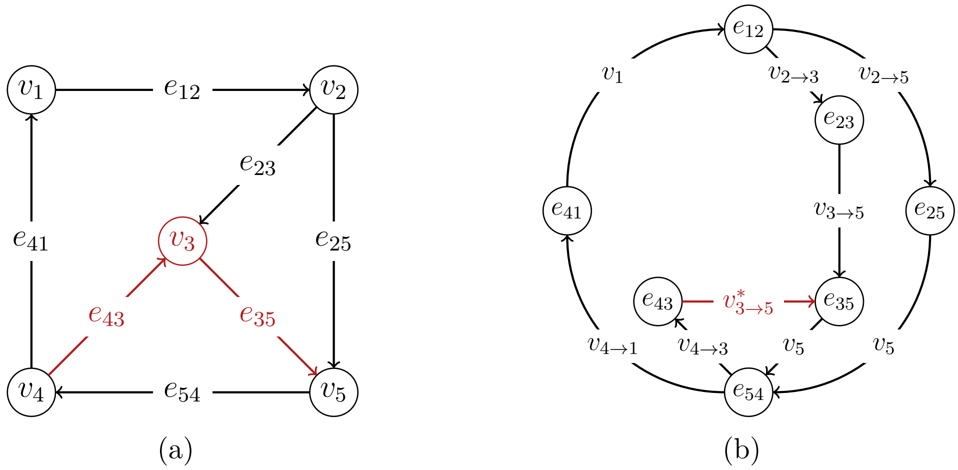


Figure 3.10: An example Directed Graph (a) transformed into its dual Edge Expanded Graph (b). The Directed Graph does not allow us to restrict specific turns, *i.e.*, let's assume we want to use route $v_4 \to v_3 \to v_5$, but the turn $e_{43} \curvearrowright v_3 \curvearrowright e_{35}$ poses a safety issue and should be avoided. In this graph, we would have to remove edge e_{35} alltogether, whereas in (b) we can remove the highlighted edge $v^*_{3\to5}$ to disallow this specific turn, leaving the other turn $v_{3\to5}$ intact, as well as remain the possibility to use route $v_4 \to v_1 \to v_2 \to v_3 \to v_5$.

any information about (im-)possible turns, to prevent certain transitions between its nodes or differently weighted intersections based on their level of difficulty, and accessibility, when using a specific path to cross it. The OSRM engine lets us transform our existing Directed Graph (Figure 3.10a) into an *Edge Expanded Graph* (Figure 3.10b) — a dual graph that switches nodes and edges and thus allows us to model such turn-based restrictions (Caldwell, 1961; Winter, 2002).

A *k*-d Tree. Searching by position for a specific instance in a very large point set, or for the nearest one of these points, is a task that has seen lots of research, as it appears in many computer related research domains, especially computer vision and computer graphics. In order to accelerate our route generation process, we integrate a k-d tree. This is a space partitioning (binary) tree and in our case d is two (*cf*. Figure 3.11). The k-d tree splits space in an alternating manner, *i.e.*, into north-south and east-west hyperplanes for each inserted point (*cf*. Figure 3.11a), where adding a single point into a balanced tree takes $\mathcal{O}(\log n)$ and building a complete tree takes $\mathcal{O}(n \log^2 n)$, if an $\mathcal{O}(n \log n)$ sorting algorithm was used to find the median at each level during construction, which also could be improved using other (pre-)sorting strategies. Once created, the k-d tree (*cf*. Figure 3.11b) supports *very* efficient range search and nearest neighbor acquisition, due to the fact that a balanced k-d tree will divide the search range of each domain, *i.e.*, longitude or latitude, into half at each

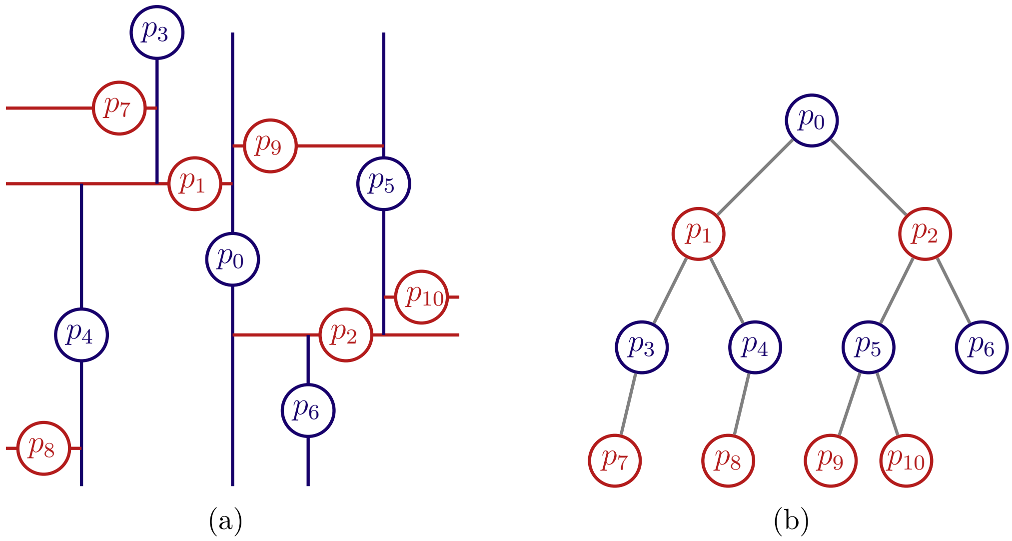


Figure 3.11: A two-dimensional space partitioning k-d tree (a) of example points and its binary k-d tree (b) that allows for very efficient range searches in a large sets of points.

level, where the considered space gets smaller with each tree level. When inserting N points into the tree, the resulting upper bound for the range search is:

$$\mathcal{T}_{RangeSearch} = \mathcal{O}(k * N^{1-\frac{1}{k}}) \quad . \tag{3.2}$$

Our Routing Approach. While a long street segment, as used by OSRM, might contain only very few points, the same distance, when using our shoreline level of detail approach, will already contain a ten-fold or more increase in points that have to be considered. As this will *significantly* increase the amount of points to be processed in each step of the routing algorithm, making it computationally very expensive, we use the k-d tree as our main data structure, in order to alleviate these issues using its very efficient range search. It allows us to effectively select and consider only pedestrian walkways and shorelines within a route segment's immediate vicinity. Based on a route from OSRM, we can calculate relevant pedestrian walkways and shorelines, *i.e.*, building facades, that we must consider for our shoreline level routing. We then connect these shorelines on each pedestrian walkway side, joining adjacent facades and bridging small gaps, *i.e.*, driveways or road crossings. This roadside aware route generation process prevents unnecessary roadside changes in dangerous places, as shoreline candidates on the other side of the street will usually be much further away than those found on the same side. Moreover, we also include a penalty for crossing the street, *i.e.*, changing pedestrian walkway sides, at an uncontrolled crossing, while Accessible Pedestrian Signals (APS) are preferred.

Pedestrian Walkway Prioritization ($\mathcal{R}_{\textbf{Walkway}}$). OSRM mostly considers road networks, but will also route along pedestrian walkways, if available. However, these are only rarely integrated, due to their limited availability and interconnection with the road. Furthermore, as OSRM considers a segment's travel speed, pedestrian walkways are used much less frequently. We therefore decrease the maximum speed on all roads when we transform map data into the edge expanded graph and modify their weights accordingly ($\mathcal{W}_{\mathcal{R}}$), leaving the pedestrian walkways unchanged ($\mathcal{W}_{\mathcal{W}}$). We similarly modify bicycle lanes to discourage their usage, however we have to differentiate between separated and combined cyclist lanes, since we would prefer a combined pedestrian/cyclist lane over using road surface.

Accessible Pedestrian Signal (APS) Prioritization ($\mathcal{R}_{\textbf{APS}}$). Another important aspect to consider for a safe routing is pedestrian crossings. These vary from uncontrolled crossings to traffic lights that have been made accessible using acoustic and haptic signals, as well as pilot tones — Accessible Pedestrian Signal (APS). Ideally, APSs also include relief symbols and exhibit long walk light phases that allow for a safe crossing. Naturally, accessible controlled crossings are to be preferred over uncontrolled ones (Matthews et al., 2014). Our preferred usage is therefore weighted as follows: APS with acoustic pilot tones ($\mathcal{W}_{\mathcal{PAPS}}$), APS with haptic signals only ($\mathcal{W}_{\mathcal{HAPS}}$), inaccessible pedestrian signals ($\mathcal{W}_{\mathcal{PS}}$), zebra crossings ($\mathcal{W}_{\mathcal{Z}}$) and finally uncontrolled crossings ($\mathcal{W}_{\mathcal{C}}$) — the latter are to be avoided at all costs. The final route always depends on the specific chosen parameters, *i.e.*, weight penalties for less accessible crossing types, and might thus change dramatically (*cf*. Figure 3.12). While this sometimes seems unintuitive to the observer, the safety benefits gained are always clearly measurable. It is important to note that this second aspect can only be considered in combination with the pedestrian walkway-based routing, as it is impossible to include pedestrian crossings into the route when we are routing on the road surface in the first place, *i.e.*, pedestrian crossings would never be considered.

Shoreline Prioritization ($\mathcal{R}_{\textbf{Shoreline}}$). One of our major motivations were the Orientation&Mobility White Cane techniques for close range guidance, as well as their complete neglect by navigation systems. This leads to problematic situations, *e.g.*, a pedestrian navigation system might ask to cross a street not far away from a controlled crossing, or to transverse an open space where there are no shorelines available at all, which conflicts with O&M techniques. We noticed in discussions with O&M trainers and during accompanied training sessions that shorelining, *i.e.*,

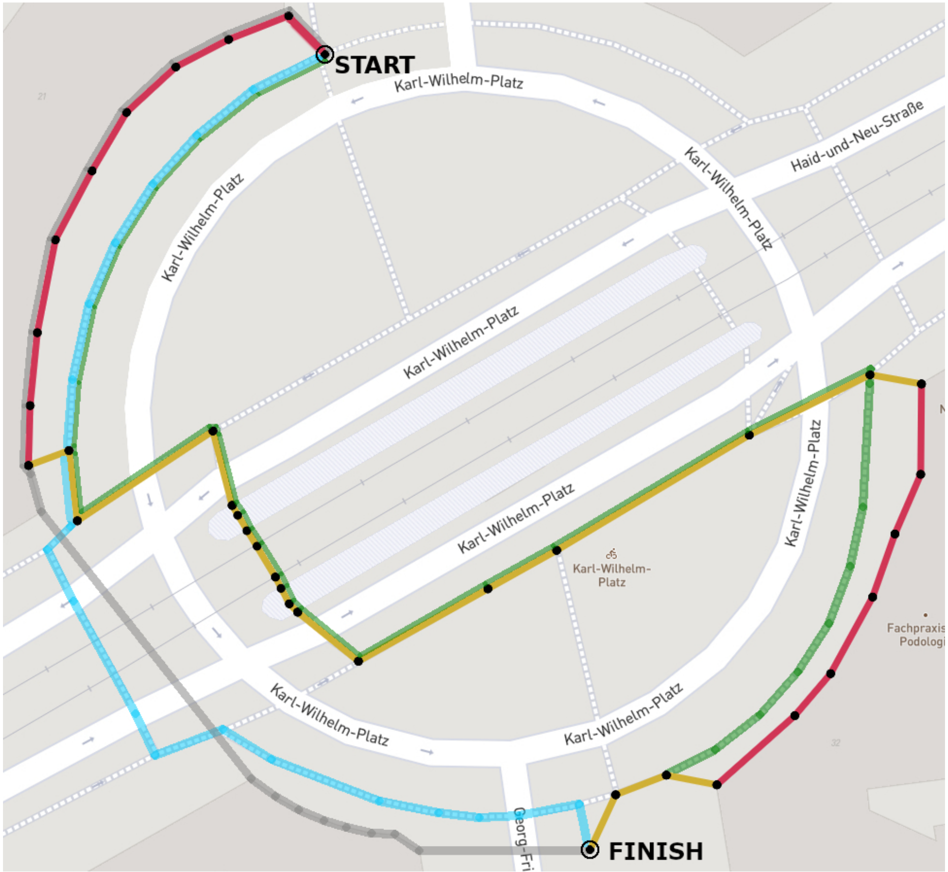


Figure 3.12: Different routing prioritizations on a challenging throughabout. We compare our walkway prioritization (**blue**), our APS prioritization (**green**), our shoreline prioritization (**gray**), and our final route (**red**/**orange**), which consists of real (**red**) and virtual (**orange**) shorelines and walkways. Please note that the lowermost and uppermost streets do not provide a controlled pedestrian crossing. Map Data © OpenStreetMap contributors.

the process of following along a shoreline for an extended period of time, was always considered an intuitive and adequate approach by all participating parties. Especially inner shorelines, *i.e.*, those that are further away from street traffic, *e.g.*, building facades or fences, were regarded as very safe and easily distinguishable when relying on the White Cane. In urban areas, such facades are directly adjacent to the pedestrian walkway and are also considered the safer alternative compared to outer shorelines, *i.e.*, curblines. Thus we also integrate available shoreline information into our routing process ($\mathcal{W}_{\mathcal{S}}$), but have to distinguish between *real* shorelines, *i.e.*, natural features such as curblines or building facades, and *virtual* shorelines, *i.e.*, when we create a virtual link between two natural features (*cf*. Figure 3.12). It allows us to generate very fine-grained routing instructions, based on shorelines as the preferred connection between two points, and to closely integrate this O&M technique into our system.

Algorithm 1: Dijkstra Algorithm – Final Route Generation

1: $d := \{0, \infty, \ldots, \infty\}$ ▷ initialize cumulative distance to nodes
2: $prev := \{0, -1, \ldots, -1\}$ ▷ track shortest path predecessor for each node
3: $q := \{(\mathcal{P}_{\mathcal{S}}, 0)\}$ ▷ priority queue by distance
4: **while** $q \neq \emptyset$ **do**
5: $\quad \mathcal{P}_u := q.pop()$
6: $\quad$ **for all** $\mathcal{L}_i \in \{\mathcal{S} \cup \mathcal{G}\}$ **do** ▷ consider $\mathcal{S}$horelines and $\mathcal{G}$eneral segments
7: $\quad\quad \mathcal{P}_v := f_{nearest}(\mathcal{P}_u, \mathcal{L}_i)$ ▷ get closest point for $\mathcal{P}_u$ w.r.t. $\mathcal{L}_i$
8: $\quad\quad$ **if** $d[u] + \Delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) < d[v]$ **then** ▷ check cost (3.3) for next point
9: $\quad\quad\quad d[v] := d[u] + \Delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i)$ ▷ update data structures
10: $\quad\quad\quad prev[v] := u$
11: $\quad\quad\quad q.push(\mathcal{P}_v, d[v])$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: **end while**

Final Route Generation Algorithm ($\mathcal{R}_{\mathbf{Final}}$). Our final proposed algorithm has to deal with situations where no building facades or pedestrian walkways are available, and thus fall back onto road-based routing, a process we would like to defer as long as possible, but that is sometimes necessary in order to generate a complete and reasonable route without any gaps or major defects in it. Figure 3.12 shows such a fine-grained route, as well as the intermediate routes created by the individual prioritizations described in the last paragraphs. Also visible is the issue that a pure shoreline routing approach exhibits a behavior that strongly contradicts O&M techniques, *i.e.*, it ignores pedestrian crossings and directly connects building facades. To create such a final route, we merge all three specialized prioritizations into a single algorithm: the pedestrian walkway ($\mathcal{R}_{\text{Walkway}}$), APS-based ($\mathcal{R}_{\text{APS}}$), and shoreline level routing ($\mathcal{R}_{\text{Shoreline}}$), and modify the Dijkstra algorithm (*cf*. Algorithm 1) in lines 6 and 7. Lines 1–3 initialize required data structures, *i.e.*, the cumulative distance to each node, the preceding node that is part of the shortest route towards each encountered node, and a distance sorted priority queue with our start point. Lines 4–5 then check this queue, as long as it is not empty, and always process the closest node, *i.e.*, $\mathcal{P}_u$. Line 6 was modified to consider $\mathcal{S}$horelines and $\mathcal{G}$eneral segments and nodes ($\mathcal{G} = \{\mathcal{W} \cup \mathcal{R} \cup \mathcal{O}\}$), *i.e.*, $\mathcal{W}$alkways, $\mathcal{R}$oads, and $\mathcal{O}$thers, *e.g.*, pedestrian traffic light nodes. We added line 7 to retrieve the nearest point $\mathcal{P}_v$ of a given segment, *i.e.*, it could be a road start, a piece of a building facade, or a walkway section. Lines 8–11 then update the internal data structures, *iff* we have found a shorter connection between our current position and the node we are currently processing.

We also use a customized cost function $\Delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i)$ in lines 8 and 9, that determines the cost to integrate a given segment into our route, based on the graph's weights $\mathcal{W}_*$, and also depends on the specific type of each individual segment:

$$\Delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) := \begin{cases} \delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i), & \delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) \neq 0 \\ \mathcal{W}_{\mathcal{S}} \cdot \|\mathcal{P}_u - \mathcal{P}_v\|_2\,, & \delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) = 0 \wedge \mathcal{L}_i \in \mathcal{S} \\ \mathcal{W}_{\mathcal{G}}(\mathcal{L}_i) \cdot \|\mathcal{P}_u - \mathcal{P}_v\|_2\,, & \delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) = 0 \wedge \mathcal{L}_i \in \mathcal{G} \end{cases} . \tag{3.3}$$

This cost function first calculates the cost to reach the next segment as:

$$\delta(\mathcal{P}_u, \mathcal{P}_v, \mathcal{L}_i) := \|\mathcal{P}_u \mathcal{L}_i\|_2 \cdot (1 + |\mathcal{C}_{\mathcal{P}_u \mathcal{P}_v}| \cdot \mathcal{W}_R) \quad , \tag{3.4}$$

where it checks, whether our currently processed node $\mathcal{P}_u$ is already a part of, and thus lies on, the inspected segment, *i.e.*, $\|\mathcal{P}_u \mathcal{L}_i\|_2 = 0$. If it is not, it furthermore determines the number of streets that have to be crossed ($|\mathcal{C}_{\mathcal{P}_u \mathcal{P}_v}|$) in order to reach $\mathcal{P}_v$, *i.e.*, it penalizes street crossings. If there are any found, their number is multiplied with the graph's weight $\mathcal{W}_{\mathcal{R}}$ of the segment between $\mathcal{P}_u$ and $\mathcal{P}_v$. This yields the final cost to reach the currently processed segment. The second case of the cost function in equation (3.3) handles instances, where our currently processed node $\mathcal{P}_u$ is already part of a shoreline, *i.e.* its distance is 0 and $\mathcal{L}_i \in \mathcal{S}$. Then we can simply multiply the distance between $\mathcal{P}_u$ and $\mathcal{P}_v$ with the shoreline segment's travel weight. Please note that in our case, real shorelines, *i.e.*, building facades, should never cross any roads, unless there exists an issue with the mapping data. For our final route, we consider all route segments that are not real shorelines or other OSM segments to become virtual shorelines (*cf*. Figure 3.12). The last case of equation (3.3) deals with all other remaining cases, *i.e.*, general segments that are made up of roads or walkways and any other type of node found in the map data. This includes pedestrian crossings, and the final cost for these nodes depends on their specific type:

$$\mathcal{W}_{\mathcal{G}}(\mathcal{L}_i) := \begin{cases} \mathcal{W}_{\mathcal{C}}, & \mathit{InformalCrossing}(\mathcal{L}_i) \\ \mathcal{W}_{\mathcal{Z}} \cdot \mathcal{W}_{\mathcal{G}}, & \mathit{ZebraCrossing}(\mathcal{L}_i) \\ \mathcal{W}_{\mathcal{PS}} \cdot \mathcal{W}_{\mathcal{G}}, & \mathit{PedestrianSignal}(\mathcal{L}_i) \\ \mathcal{W}_{\mathcal{HAPS}} \cdot \mathcal{W}_{\mathcal{G}}, & \mathit{HapticSignalAPS}(\mathcal{L}_i) \\ \mathcal{W}_{\mathcal{PAPS}} \cdot \mathcal{W}_{\mathcal{G}}, & \mathit{PilotToneAPS}(\mathcal{L}_i) \\ \mathcal{W}_{\mathcal{G}}, & \mathit{otherwise} \end{cases} , \tag{3.5}$$

where the $\mathcal{W}_* \in \mathbb{R}_+$ are the corresponding penalties for each type of crossing, as defined by the APS prioritization, and $\mathcal{W}_\mathcal{G}$ is the segment's or node's general travel weight from the routing graph. We multiply these individual penalty weights — except for the informal crossing, as it is not contained in the routing graph and thus does not have a weight associated to it — with their general routing graph weights to additionally account for contained segment information. Otherwise we might lose relevant information, *e.g.*, the possible speed on a segment, as OSM data often contains the pavement conditions, *i.e.*, (fine) gravel, pebblestones, cobblestones, mud, sand, concrete, or others. Also, segments that are marked as temporarily inaccessible, *e.g.*, due to ongoing construction or other issues, would be misused in this case.

Choosing Weights. Finally, we integrate our assumptions about previously analyzed crossing preferences into our model, ensuring for all relevant pedestrian crossing penalty weights $\mathcal{W}_*$:

$$\mathcal{W}_\mathcal{C} \gg \mathcal{W}_\mathcal{Z} > \mathcal{W}_{\mathcal{PS}} > \mathcal{W}_{\mathcal{HAPS}} > \mathcal{W}_{\mathcal{PAPS}} \gg 1 \quad . \tag{3.6}$$

Furthermore, for our routing algorithm to prefer available shorelines over general road or walkway segments as well as any other encountered types, we also ensure:

$$\mathcal{W}_\mathcal{G} > \mathcal{W}_\mathcal{R} > \mathcal{W}_\mathcal{W} > \mathcal{W}_\mathcal{S} \quad . \tag{3.7}$$

For our performed evaluations, we use the following penalty values for pedestrian crossings: $\mathcal{W}_\mathcal{C} = 600$, $\mathcal{W}_\mathcal{Z} = 30$, $\mathcal{W}_{\mathcal{PS}} = 20$, $\mathcal{W}_{\mathcal{HAPS}} = 10$, and $\mathcal{W}_{\mathcal{PAPS}} = 5$. Additionally, when constructing the Edge Expanded Graph (*cf*. Figure 3.10), we modify the road network speeds as follows: The general walking speed on pedestrian walkways and any other traversable segments is set to 5 km/h, and is multiplied with 0.5 for roads, 0.9 for combined bicycle lane segments, 0.95 for separated bicycle lane segments, and 1.5 for building facades, *i.e.*, inner shorelines. We mostly ignore highways, railways, segments that are marked as under construction, dedicated bicycle lanes, and other segment types that should similarly be discouraged, setting their speed to zero. These speeds are then implicitly used by the routing algorithm, as a segment's length divided by its corresponding speed yields the travel time for that segment, *i.e.*, its weight used for the cost function. This allows the routing algorithm to prefer sufficiently shorter segments, even if less accessible, but for the same length it will prefer the most accessible one according to the pre-defined criteria.

3.2.3 Evaluation

As we have previously identified concrete criteria to evaluate our algorithmic approaches against, we separate this process into three steps: First, use purely random routes of varying length, second, rely on public transit stations as start/end points, and finally, perform a small Wizard of Oz experiment with a single blind participant, as there was no GNSS receiver available to us with a sufficient positioning precision. Therefore, we prepare our map data as previously described, which is computationally very expensive due to the huge amount of map nodes that require processing when we consider individual shoreline segments. However, it has to be performed only once for each area, the succeeding route generation costs much less per route due to us using the k-d tree data structure, and largely depends on its overall length.

Randomly Generated Routes

Our different route prioritizations are initially evaluated against an OSRM routing algorithm that already integrates pedestrian walkways, on 1000 randomly created, mostly shorter routes, *i.e.*, less than 1000 meters. We conducted this first experiment to check what kind of route changes the individual routing prioritizations (*cf.* Section 3.2.2) would introduce and whether these were individually feasible. We did this to detect if they would possibly create completely unreasonable routes or route against their intended purpose, *e.g.*, due to incorrectly chosen weights or penalties.

The random evaluation took place in a large municipality, which includes multiple cities as well as rural communities, where OSM data was rather scarce, but we included it in order to see whether our proposed prioritizations could also be applied in such areas, not just in urban city centers, where OSM data availability is usually much higher. We use 1000 almost random points in this area as start points. Their only constraint is that they must lie within two meters of a random road or walkway segment, as this models real world usage, where people would almost never manage to start a routing request exactly on an OSM node's location, as well as from within a building. The end points of these random routes must be on a road that is up to 700 meters away — straight beeline distance, where on a typical regularly squared city grid a diagonal distance of 700 meters yields an average travel distance of almost 1000 meters, according to the Pythagorean theorem — at least 100 meters away,

Table 3.3: We compare $\mathcal{R}_{\text{OSRM}}$ (includes pedestrian walkways) with our $\mathcal{R}_{\text{Walkway}}$, $\mathcal{R}_{\text{APS}}$, and $\mathcal{R}_{\text{Shoreline}}$ prioritizations, as well as our $\mathcal{R}_{\text{Final}}$ route. $\bar{\mathcal{D}}_{\text{travel}}$ is the average travel distance in meters, $\mathcal{W}_{\text{walkway}}$ the route's percentage of pedestrian walkways, $\mathcal{P}_{\text{signal}}$ the percentage of used pedestrian signals and $\mathcal{C}_{\text{informal}}$ of informal crossings, and finally $\mathcal{S}_{\text{real}}/\mathcal{S}_{\text{virtual}}$ the percentages of real/virtual shorelines. We also include relative changes to $\mathcal{R}_{\text{OSRM}}$ ($\pm$%), "—" not considered routing aspects, and "(...)" denotes absolute numbers that cannot be compared to $\mathcal{R}_{\text{OSRM}}$ (please refer to route discussions for further details on these issues).

	$\bar{\mathcal{D}}_{\text{travel}}$	$\mathcal{W}_{\text{walkway}}$	$\mathcal{P}_{\text{signal}}$	$\mathcal{C}_{\text{informal}}$	$\mathcal{S}_{\text{real}}$	$\mathcal{S}_{\text{virtual}}$
$\mathcal{R}_{\text{OSRM}}$	**458**	12.8	71.0	29.0	—	—
$\mathcal{R}_{\text{Walkway}}$	466 +1.7%	**16.4**+28.1%	**83.9**+18.2%	**16.1**-44.5%	—	—
$\mathcal{R}_{\text{APS}}$	**464**+1.3%	**16.4**+28.1%	**92.5**+30.3%	**7.5** -74.1%	—	—
$\mathcal{R}_{\text{Shoreline}}$	(139)	—	43.7 -38.5%	56.3 +94.1%	**24.5**	**12.3**
$\mathcal{R}_{\text{Final}}$	(162)	**17.6**+37.5%	57.3 -19.3%	42.7 +47.3%	**20.9**-14.7%	**11.0**-10.6%

and, similarly to the start point, within two meters of a road or walkway segment. Due to technical limitations, *i.e.*, memory and processing time for the significantly increased number of points when considering shorelines, we could only evaluate our shoreline and final routing approach, which integrates shorelines as well, on 400 of the aforementioned 1000 routes, *i.e.*, mostly shorter ones. Thus, absolute numbers, *i.e.*, travel distances, are not directly comparable, whereas all relative percentages on included walkways, pedestrian signals, informal crossings, and shorelines, remain so. Finally, we evaluate our different prioritizations according to our pre-defined criteria, *i.e.*, average travel distance ($\bar{\mathcal{D}}_{\text{travel}}$), percentage of pedestrian walkways ($\mathcal{W}_{\text{walkway}}$), percentage of pedestrian signals ($\mathcal{P}_{\text{signal}}$), percentage of informal pedestrian crossings ($\mathcal{C}_{\text{informal}}$), and shoreline integration ($\mathcal{S}_{\text{real}}/\mathcal{S}_{\text{virtual}}$), where applicable.

Pedestrian Walkway Prioritization ($\mathcal{R}_{\text{Walkway}}$). Our first random evaluation results in table 3.3 indicate that our walkway prioritization only slightly increases the average travel distance — less than 2% — making the routes comparable *per se*, as they are of very similar length. This prioritization is able to increase the percentage of taken walkways (+28%), even as $\mathcal{R}_{\text{OSRM}}$ already considers these, they are just not preferred over roads. We investigated this issue and found that OSM does not contain sufficient walkway data, or we could possibly have seen an even greater improvement. Nonetheless, it already integrates more pedestrian signals (+18%), and reduces the amount of informal crossings. As shorelines are never considered in this prioritization, we cannot provide any numbers about their usage, same as for $\mathcal{R}_{\text{OSRM}}$.

Accessible Pedestrian Signal (APS) Prioritization ($\mathcal{R}_{\mathbf{APS}}$). Table 3.3 furthermore indicates that $\mathcal{R}_{\text{APS}}$ can improve over $\mathcal{R}_{\text{Walkway}}$, as it also has to integrate walkways in the same way (*cf*. Section 3.2.1), but prefers accessible pedestrian crossings whenever possible. Its main difference to $\mathcal{R}_{\text{Walkway}}$ is to ensure that required pedestrian crossings follow our accessibility criteria, *i.e.*, prefer APS over informal crossings, which is already supported by our first evaluation, as it more than halves the amount of informal crossings, *i.e.*, from 16.1% to 7.5%, and also manages to increase the amount of included APS, *i.e.*, +30.3% *vs.* +18.2% for $\mathcal{R}_{\text{Walkway}}$. While it also manages to only insignificantly increase the travel distance, it too does not yet consider shorelines, similar to $\mathcal{R}_{\text{Walkway}}$ and $\mathcal{R}_{\text{OSRM}}$.

Shoreline Prioritization ($\mathcal{R}_{\mathbf{Shoreline}}$). Our $\mathcal{R}_{\text{Shoreline}}$ prioritization generates its routes quite differently, as show in table 3.3. It does not consider pedestrian walkways ($\mathcal{W}_{\text{walkway}}$), as it integrates the additional features of real and virtual shorelines, which none of the other prioritizations support. However, it only uses pedestrian walk light signals (−39% $\mathcal{P}_{\text{signal}}$) when there are no shorelines around, *i.e.*, it's often routing on roads or walkways in rural areas. Then, it instead prefers to use informal crossings (+94% $\mathcal{C}_{\text{informal}}$), *i.e.*, it directly connects available real shorelines by creating virtual ones, even when crossing a street, although such behavior is not realistic for real world usage anyway (*cf*. the gray line in Figure 3.12). Due to technical limitations, we could only compare $\mathcal{R}_{\text{Shoreline}}$ on 400 of the 1000 routes, mostly shorter ones, thus the average distance has decreased significantly and is not directly comparable.

Final Route ($\mathcal{R}_{\mathbf{Final}}$). As $\mathcal{R}_{\text{Final}}$ also integrates shorelines (*cf*. $\mathcal{R}_{\text{Shoreline}}$), its average distance is reduced in the same way and thus not comparable. Please also note that here the relative changes for real and virtual shoreline usage are not based on $\mathcal{R}_{\text{OSRM}}$, but are instead compared to $\mathcal{R}_{\text{Shoreline}}$'s values. Regarding shoreline integration, both approaches only differ by up to 15%, which shows us that the shoreline prioritization provides a real measurable benefit, as the proposed sections are almost completely considered in the final route generation process. Similarly, $\mathcal{R}_{\text{Walkway}}$'s suggested improvements for walkway integration (+28%) are not only nicely integrated, but even improved upon by $\mathcal{R}_{\text{Final}}$, *i.e.*, the final route contains +37% more walkways than $\mathcal{R}_{\text{OSRM}}$. Only the amount of $\mathcal{P}_{\text{signal}}$ (APS) decreases significantly, *i.e.*, −20%, however trade-offs between individual routing prioritizations were to be expected. Nonetheless, any perceived trade-offs should be further investigated and might be mitigated in the future, using carefully adapted weights and penalty values.

Table 3.4: Public transit station-based results for 1870 routes, *i.e.*, ten semi-random routes for each of the 187 public transit stations available in the city of Karlsruhe. Please refer to table 3.3 for further details about compared $\mathcal{R}_{\text{outes}}$ as well as the evaluated aspects.

	$\bar{\mathcal{D}}_{\text{travel}}$	$\mathcal{W}_{\text{walkway}}$	$\mathcal{P}_{\text{signal}}$	$\mathcal{C}_{\text{informal}}$	$\mathcal{S}_{\text{real}}$	$\mathcal{S}_{\text{virtual}}$
$\mathcal{R}_{\text{OSRM}}$	**621**	26.4	**76.8**	**23.2**	—	—
$\mathcal{R}_{\text{Walkway}}$	**654** +5.3%	**46.4**+75.8%	76.3 -0.7%	23.7 +2.2%	—	—
$\mathcal{R}_{\text{APS}}$	655 +5.5%	**45.5**+72.3%	**86.4**+12.5%	**13.6**-50.0%	—	—
$\mathcal{R}_{\text{Shoreline}}$	(178)	—	44.2 -42.4%	55.8 +141%	**31.4**	**7.9**
$\mathcal{R}_{\text{Final}}$	(198)	22.0 -16.7%	64.6 -15.9%	35.4 +52.6%	**26.0**-17.2%	**8.2**+3.8%

Public Transit Station-Based Routes

The random routes were used to investigate the suitability of each prioritization and their effects on the overall route. However, these routes do not represent the actual movement patterns of people living in urban areas at all. The most interesting routes to be improved for people with visual impairments would also be their most common ones, *e.g.*, towards home or the workplace, visiting friends, meeting at specific sights, businesses or public places. Furthermore, large distances are only rarely covered by foot, but usually by means of public transport, *i.e.*, buses and tram lines.

Therefore, after consulting with people with visual impairments as well as O&M trainers, we investigate public transit station-based routes, *i.e.*, to start or end at such a public transit station. Luckily, these are also part of the OSM data and create a semi-random distribution over a city's area, as such stations are usually placed neither to close to each other, nor to far away. It should thus be considered that this evaluation takes place in an urban area only (*cf*. Figure 3.9), where OSM data availability and quality is much higher than in most areas of the random route evaluation, as there is a quite active volunteer community in the city of Karlsruhe. There are 187 available public transit station locations and we calculate ten semi-random routes per instance, where we use the same limitations as for the random route evaluation, *i.e.*, we route towards a random point, within two meters of a road or walkway that is at most 700 meters away. However, we also have the same limitations for shorelines as with the random route evaluation, *i.e.*, we cannot compare the average distances, as we had to reduce the maximum distance of 700 meters down to 250 meters, but are able to compare all other evaluated aspects, where applicable.

Pedestrian Walkway Prioritization ($\mathcal{R}_{\text{Walkway}}$). Similar to the random route-based evaluation, the overall travel distance does not change significantly, *i.e.* +5% (*cf*. Table 3.4), however, here the amount of integrated walkways is significantly increased at +76%. Moreover, the amount of utilized APSs and informal crossings is almost unchanged. This is in line with our previous observation of an increased pedestrian walkway data availability in urban areas — $\mathcal{R}_{\text{Walkway}}$ is very similar to $\mathcal{R}_{\text{OSRM}}$ in all other aspects, but much better integrates walkways, if they are available. As for the random route evaluation, shorelines are not considered by $\mathcal{R}_{\text{Walkway}}$.

Accessible Pedestrian Signal (APS) Prioritization ($\mathcal{R}_{\text{APS}}$). Table 3.4 also indicates that while the APS prioritization exhibits the largest travel distance (+5.5%), it's only minor compared to the next best one, *i.e.*, $\mathcal{R}_{\text{Walkway}}$ at +5.3%, and still insignificant compared to $\mathcal{R}_{\text{OSRM}}$. The amount of $\mathcal{W}_{\text{walkway}}$ also stays almost the same. Most importantly, $\mathcal{P}_{\text{signal}}$ increases by +12.5% from $\mathcal{R}_{\text{OSRM}}$, and even a tiny bit more for $\mathcal{R}_{\text{Walkway}}$. This shows that $\mathcal{R}_{\text{APS}}$ performs as intended, as it leaves the $\mathcal{W}_{\text{walkway}}$ almost unchanged, while significantly increasing the integration of pedestrian walk light signals. Finally, the two prioritizations show that they can already contribute individually significant improvements when compared to $\mathcal{R}_{\text{OSRM}}$.

Shorelines Prioritization ($\mathcal{R}_{\text{Shoreline}}$). Again, $\mathcal{R}_{\text{Shoreline}}$ does not consider pedestrian walkways, significantly increases $\mathcal{C}_{\text{informal}}$ (+52.6%), and benefits from a better shoreline availability, *i.e.*, $\mathcal{S}_{\text{real}}$ is 31.4% (for random routes it was only 24.5%). Also, the amount of required virtual shorelines is lower for this setting, *i.e.*, 7.9% *vs*. 12.3% for the random route evaluation, mostly due to a significant increase in available pedestrian walkways, which requires less, and also shorter, virtual segments to connect these. We had to limit the maximum route length to 250 meters in this evaluation, for similar reasons as before, rendering the average distance incomparable.

Final Route ($\mathcal{R}_{\text{Final}}$). We again compare $\mathcal{R}_{\text{Final}}$'s differences to $\mathcal{R}_{\text{OSRM}}$ as well as our shoreline-based prioritization in table 3.4. Similar to the random route evaluation, integrated shorelines proposed by $\mathcal{R}_{\text{Shoreline}}$ are mostly kept in the final routing process, while $\mathcal{P}_{\text{signal}}$ is reduced by a similar amount, *i.e.*, −16%. Interestingly, the amount of integrated pedestrian walkway has decreased by −16.7%, but this is easily explained by the much better OSM data availability, as in urban areas, there are not only much more pedestrian walkway instances labeled, but building facade nodes next to these as well. Thus, $\mathcal{R}_{\text{Final}}$ integrates more available shorelines at the cost of using less pedestrian walkways, again a trade-off that was to be expected.

Figure 3.13: These example directions are constructed from our generated route and exhibit a great level of detail as well as especially relevant information for Orientation&Mobility's shorelining: (0) *"Please turn north until you reach a building."*, (1) *"Follow the facade to the left for 6 meters."*, (2) *"Continue at 1 o'clock for 14 meters to cross a driveway."*, (3) *"Follow the facade for 13m."*, ..., (11) *'Continue at 10 o'clock for 5 meters."*, (12) *"You have reached your destination."* Map data © OpenStreetMap contributors.

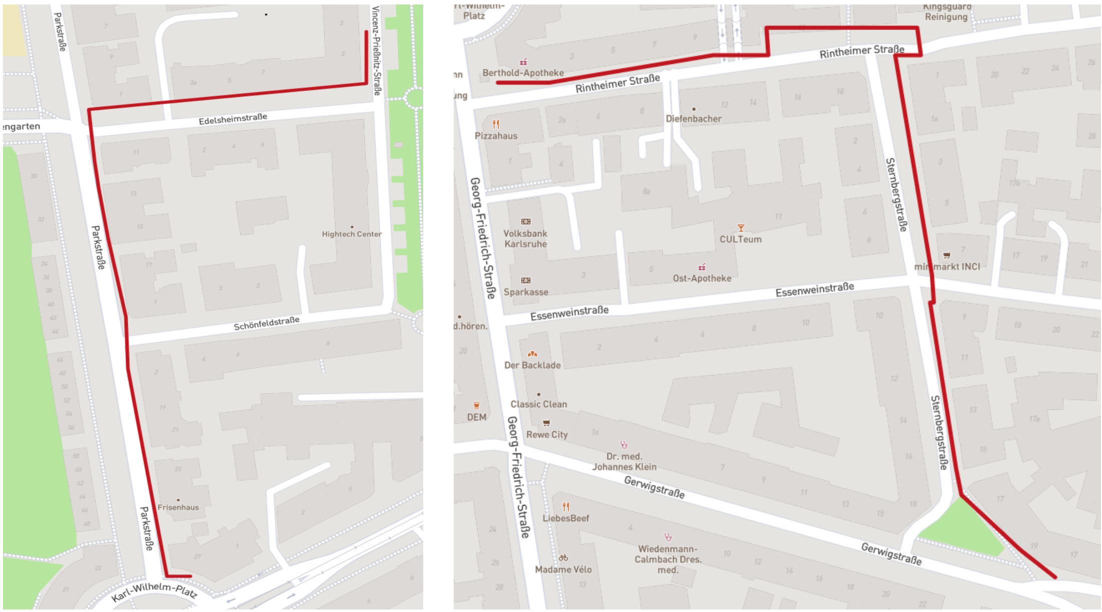
Directions

Our final algorithm allows us to generate very fine-grained route segments, *i.e.*, down to an individual shoreline level (*cf*. Figure 3.13) as we had originally anticipated. To generate directions of this low-level granularity, we compare an upcoming segment's relative location to the last considered position, in order to provide relative directions as well as the travel distance towards the next location. Furthermore, we utilize available relevant OSM data, *i.e.*, a segments specific type, *e.g.*, whether it is a real or virtual shoreline, a walkway, or a road. Finally, we integrate information about upcoming nodes, *e.g.*, an APS, an informal pedestrian crossing or a driveway crossing between building facades, to provide relevant spatial information of the route ahead.

Wizard of Oz

Due to the unavailability of GNSS receivers that provide sufficiently precise positioning information, *i.e.*, less than 1m in urban areas, we could not test our proposed shoreline level of routing approach under real conditions. Instead we had to rely on a Wizard of Oz type of user experiment (Dahlbäck et al., 1993; Kelley, 1983, 1984), where we created the navigation instructions for two 400m long routes beforehand — generated by our impairment aware routing algorithm — and manually read these out to the single participant, who had been blind since birth.[9]

[9] Many thanks to Gerhard Jaworek for participating in this experiment and his valuable feedback.



(a) (b)

Figure 3.14: The routes for the Wizard of Oz experiment. We tested our proposed shoreline-based routing on two routes in an urban neighborhood that contained many building facades, driveways and pedestrian crossings. Map data © OpenStreetMap contributors.

Similar to our public transit-based evaluation, both routes start or end at a public transit station. They contain an informal pedestrian crossing, a zebra crossing, and multiple driveways that interrupt natural shorelines, *i.e.*, building facades (*cf.* Figure 3.14). Since one of our main motivations was to understand how shoreline aware routing could improve general navigation information in urban scenarios, the two routes mostly consist of such real and virtual shorelines. Furthermore, we investigated in a questionnaire, whether our fine-grained navigation instructions would improve the participant's spatial awareness, asking specific questions, amongst others, about his confidence during the experiment, whether he appreciates the knowledge being presented in advance, how detailed these instructions should be, how reliable he considers the instructions, whether the information increased his spatial awareness and whether he had any further suggestions to improve this approach.

The participant neither knew the two routes beforehand, nor were they in an area the participant was familiar with, to simulate a completely unknown area, usually a very stressful experience for people with visual impairments and thus preferably avoided. A supervisor would follow closely to ensure the participant's safety, interactively read the generated directions out loud at predefined points, while moving along the route, simulating the usage of purely GNSS-based routing.

Results & Suggestions. The participant greatly appreciated information about the precise location, relative direction, *e.g.*, when rounding a corner, and remaining length of a natural shoreline. Information about the next shoreline as the current one ended was found invaluable, especially the specific direction in which to continue and how far until the next shoreline would be reached. Receiving knowledge about the type of gap to cross between shorelines in advance, *e.g.*, whether it is just a driveway or a much larger road, was also very much appreciated. Furthermore, the participant liked the clock system to communicate relative directions very much (*cf.* Sánchez and de la Torre (2010)) and even mentioned that it felt very familiar due to it also being used to describe food piece locations on a plate, a happenstance we had not been made aware of before integrating it into this experiment.

All in all, the participant very much valued our provided shoreline level information, and would like to continue using it, especially for unknown terrain, but also in daily navigation scenarios on common errands. Most importantly, the participant noted a much better ability to deal with upcoming issues, *i.e.*, shoreline gaps or road crossings, due to information on how to proceed being provided in advance, increasing the spatial awareness of the following route steps. Furthermore, the participant wished for the possibility to interactively ask the navigation system about even farther away shoreline segments, when there was time for it, *e.g.*, while following a long segment, as well as additional information about an intersection's layout. The participant also suggested to create a pilot tone for virtual shoreline segments, or even use tactile feedback on the cane itself, in order to better locate the next real shoreline segment, and continue towards it in a straight line without veering of. Finally, the participant insisted that the interface must be highly configurable, especially the verbosity level and repetitiveness, but also the type of direction announcement, as some users might prefer relative degrees or north/south instructions instead of the clock system.

Although this Wizard of Oz experiment was of very limited scope, *i.e.*, a single participant on two short routes in an unknown area, it provided us with an invaluable amount of insight w.r.t. the potential usefulness of our proposed approach and additionally desired features that should be investigated further in the future. Furthermore, it partly inspired us to start working on our visual shoreline detection from a street level perspective, which we discuss in detail in section 4.2.

3.2.4 Conclusion

We have proposed a routing approach that provides measurable benefits for people with visual impairments w.r.t. accessible and impairment aware routing generation. We rely on publicly available geospatial information, however, more precise information exists within municipalities that should be integrated as well, a process which would allow us to also integrate other natural features, *i.e.*, curblines. To the best of our knowledge, our proposed approach is the first of its kind, *i.e.*, to integrate and route on such a fine-grained level of detail that is closely modeled after the specific requirements of people with visual impairments has not been done before. The statistical evaluation shows an improvement in the amount of used pedestrian walkways and accessible pedestrian crossings, while there is only a marginal increase in travel distance. Our proposed approach also yields safer routes w.r.t. our predefined criteria: it prevents uncontrolled crossings, prefers as accessible as possible pedestrian signals, and most importantly, integrates shorelines whenever possible. Finally, we have also tested two generated routes in a small Wizard of Oz experiment.

While this first experiment has shown promising results, a further, large-scale user study is still required. Generated routes still show some room for improvement: They should rely on verified geospatial information, in contrast to volunteered one that is often erroneous or incomplete, and include other types of shorelines as well, *e.g.*, curblines, tactile pavements integrated into pedestrian walkways, and ideally walkway borders in general. Finally, as is always the case when dealing with very individual disabilities, personal preferences and abilities have to be accounted for.

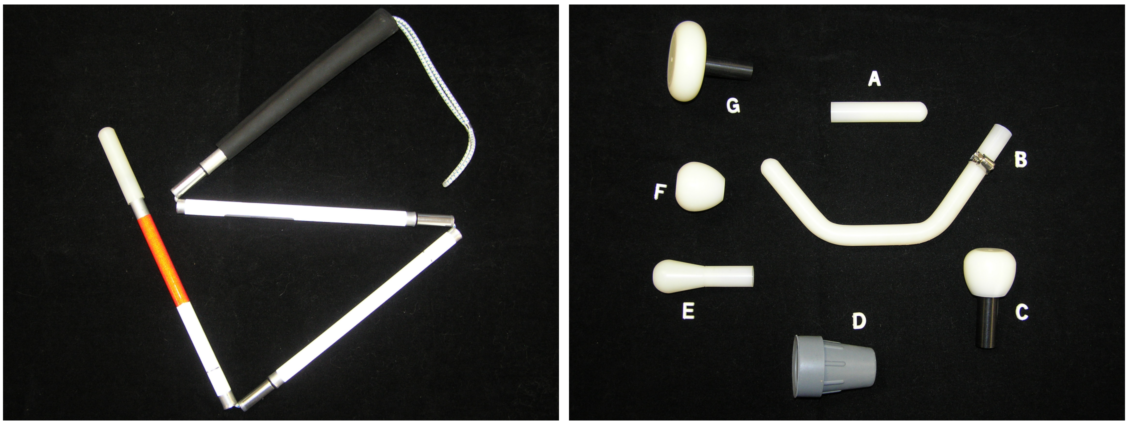
A general feasibility study should be performed, to examine the localization precision of latest generation GNSS systems, once they are publicly available, a feature we lacked for our own evaluation. However, even a promised average accuracy of only one meter would be insufficient to communicate the exact location of the next shoreline segment's beginning, or the precise location of a pedestrian signal's pushbutton, as GNSS-based positioning is usually much worse in urban city street canyons. This inspired us to further investigate computer vision techniques, to detect such shorelines (*cf*. Section 4.2) as well as assist in pedestrian crossings (*cf*. Section 4.3) in real time from a pedestrians viewpoint, without relying on GNSS information. All these systems, when taken together, assist in self-sufficient Orientation&Mobility for people with visual impairments and are potentially especially useful in unknown areas.

Chapter 4

Mobility

> “The first principle is that you must not fool yourself
> — and you are the easiest person to fool.”
>
> – Richard P. Feynman

To this day, the White Cane remains the most widely upon relied device for outdoor mobility of people with visual impairments. Its specific techniques, which make it such an effective tool, have to be learned in an elaborate process — Orientation & Mobility (O&M) training. O&M specialists, in many countries an accredited job title and often supported from the government, instruct people with visual impairments in these techniques: sensory development, indoor and outdoor cane usage, shorelining, landmark and compass usage, squaring off (a technique for crossing open spaces), using pedestrian crossings safely, analyzing traffic patterns, relying on a guide dog, using public transport, dealing with disorientation, as well as self-protective techniques, *i.e.*, trailing, and finally techniques for family and relatives, *i.e.*, being a sighted guide and voice guiding. They provide the necessary training to people with visual impairments of all ages, often in one-on-one sessions, to learn and further develop the necessary skills and concepts, whether they are infants, children or adults. Such training sessions might have to be performed across the entire life span, as sometimes already known techniques have to be relearned or people move to a different place and have to get accustomed to it. Primarily, O&M supports people with visual impairments to maneuver around safely, independently, effectively and efficiently, but O&M specialists might consult with families or (physical) therapists as well.



(a) (b)

Figure 4.1: (a) A long cane, primary mobility tool for people with visual impairments, foldable for easier storage. (b) Different cane tips: (A) Pencil (B) Bundu Basher (C) Ball Race Overfit (D) Rubber Support Cane (E) Pear (F) Rural (G) Jumbo Roller. These cane tips allow for customization depending on user preference and situation. 🅭🅯[1] (Sarah Chester)

Orientation&Mobility are two highly interconnected concepts, where orientation is mostly focused on *high-level* information about the place we are currently in (*cf*. Chapter 3). Mobility, as its complementary counterpart, considers *low-level* information about the immediate surroundings, *e.g.*, the accessible section in front, whether there are any obstacles, or the availability of shorelines, in order to teach people with visual impairments how to *roam* a place without tripping, falling, or bumping into obstacles. Furthermore, specific situations are considered and trained, such as how to cross a street — this might well be a very specific pedestrian crossing on the daily way to the workplace — or how to effectively use shorelines in dense urban areas. O&M specialists instruct the usage of observable cues to improve location awareness. These are often very specialized and have to be individualized for each trainee, depending on the urban or rural settings, personal abilities, and used assistive devices as well as their personalizations (*cf*. Figure 4.1).

Specifically in cases where O&M techniques rely on features that exhibit a distinctive visual appearance, computer vision-based techniques can provide relevant assistance (*cf*. Section 1.2). For example, Accessible Pedestrian Signals (APSs) — if traffic lights have been made accessible by communities at all — are first and foremost a visual semaphore intended for traffic flow management. This visual feature, *i.e.*, a red or green light, also walk or don't walk signals (*cf*. Figure 2.3), can be located

[1] https://en.wikipedia.org/wiki/White_cane
CC BY 3.0 https://creativecommons.org/licenses/by/3.0/

and classified using computer vision algorithms and then communicated to the user. Similarly, zebra crossings are also relying on a distinctive visual feature, *i.e.*, clearly visible white stripes as pavement markings, that provide only an almost imperceptible haptic feedback to the White Cane. While these two specific traffic related objects are rather easy to detect, due to their very unique appearance, other relevant obstructions or features are much harder to identify visually. For example, the accessible section will change its appearance based on whether we are in an urban or rural area. Additionally, there exists a multitude of different pavements for pedestrian crosswalks, road surfaces or general walkways, *e.g.*, cobblestones, small and large paving tiles, concrete, or tar with or without additives like crushed rocks or synthetic materials, to just name a few of the possible patterns and combinations, where a sufficient amount of training data would be required for all of these when using an appearance-based approach. Furthermore, obstacles come in numerous different forms in urban areas alone, *e.g.*, trash cans, lamp posts, other pedestrians, bicycles, cars, trucks, plants, construction sites and many more. All of these issues make it very hard for computer vision techniques to provide sufficient and adequate certainty. This is especially relevant, since we always have to consider safety issues. While we try to provide additional information of the kind that cannot be perceived using a White Cane or GNSS device, we primarily want to inform users as early as possible about upcoming issues, so they can include it into their decision process.

Therefore, this chapter's focus is the creation of assistive systems to support mobility, while the orientation aspects of O&M were already investigated further in chapter 3, in order to decrease the identified gap between these two (*cf*. Section 1.3). We investigate methods to assist in free roaming, *i.e.*, accessible section and obstacle detection in section 4.1. Moreover, in section 4.2 we focus our efforts on natural features that are used for O&M techniques, *i.e.*, shorelining. Finally, we investigate how to improve the accessibility of pedestrian crossings in section 4.3.

Our Contributions. We present an accessible section detection algorithm in section 4.1.2, based on depth data, to deal with the issues of varying pavement appearance. In section 4.2.2, we then propose a novel approach to precisely localize inner shorelines (Koester et al., 2018), investigate their availability, direction and length, upcoming gaps, and possible continuations. Finally, we use a deep learning approach to detect zebra crossings, traffic lights, and other relevant objects, to assist before and during the crossing process of pedestrian crossings in section 4.3.2 (Koester et al., 2019).

4.1 Accessible Section & Obstacle Detection

We consider the duality of obstacle detection and accessible section detection, as both issues pose similar problems. While it is definitely possible to detect many different obstacle classes with modern approaches, *i.e.*, deep learning, it is absolutely impossible to detect *all* of those obstacle classes one might eventually encounter. For traditional approaches, a specific feature and classifier would be required per obstacle, for deep learning the difficulty of manually creating features is instead offloaded to the elaborate process of creating a sufficient amount of training data. Furthermore, it is much harder to reason about safety aspects for the learned networks that result from deep learning approaches, than it is for hand-crafted features. We therefore only consider the problem of detecting accessible section, which implicitly allows us to detect obstacles as well — a lack of accessible section in a specific location. Our approach is based on depth data that allows us to provide certain guarantees about its reliability and working conditions. As detecting the accessible section, and this includes overhead obstacles as well, is a critical safety technique in O&M training, we have to value the user's safety as an important factor, in accordance with our own key concepts identified in section 2.5.

4.1.1 The White Cane

One of the most relied on accessibility devices for O&M training is by far the White Cane, an integral part of mobility. It is mainly used to perceive one's immediate surroundings, especially the ground plane one is currently walking on, to detect low obstacles, find small corridors or openings, follow along tactile paving, warn before drop-offs such as stairs, and many more. However, due to its physical size limit — people with visual impairments often prefer the lightest version they can find in order to prevent early exhaustion from the constant arcing motion — it has a very limited range, *i.e.*, its length is usually only up to 1.5 meters. While this is absolutely sufficient for most O&M techniques, it would be interesting to significantly enhance this range. We propose to do this using computer vision and detect the accessible section in front of the user. We already suggested an earlier approach in Koester et al. (2013), which was based on the depth reconstruction from Geiger et al. (2010). However, it was very susceptible to noise in the depth data and not real time capable. Other relevant related work is further discussed in section 2.3.1.

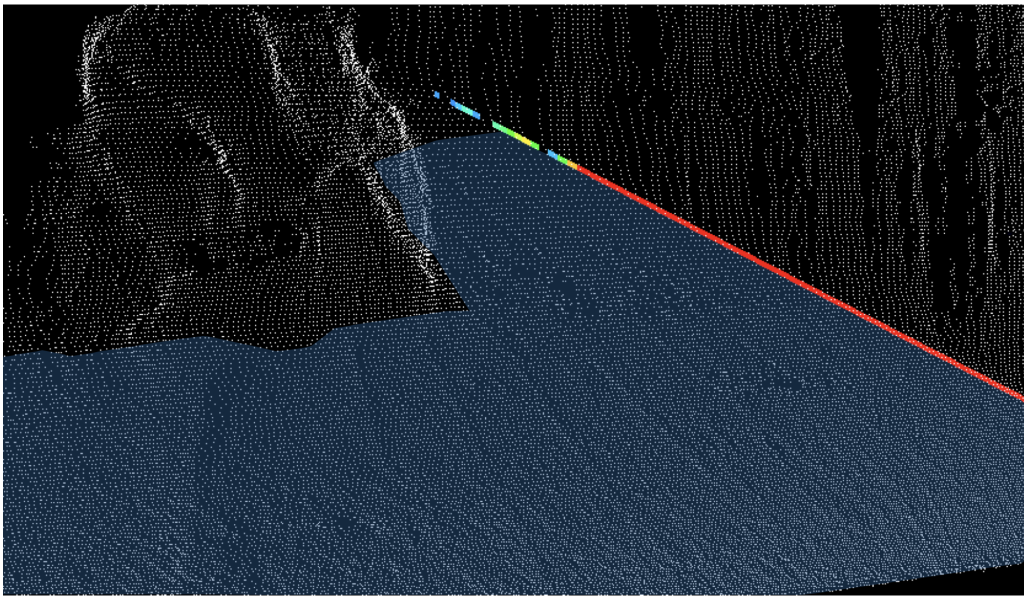


Figure 4.2: Accessible section detection example. Right of the accessible section (blue) is a building's facade, while the blob on the left is a parked car. Also shown is a detected shoreline (Section 4.2), as both approaches rely on depth data and share the same techniques.

4.1.2 Detecting the Accessible Section

We propose a more robust approach as compared to Koester et al. (2013), that utilizes RANSAC, but redefine the problem itself for improved computational efficiency and robustness. Our algorithm relies on depth data instead of disparity data used in our prior approach, *i.e.*, here we analyze point cloud data. This allows our algorithm to also be used with other sensors that provide such depth data, not just the stereo cameras we used, but also potentially computationally inexpensive Light Detection and Ranging (LIDAR) sensors that might someday become small and low powered enough, to be as usable for people with visual impairments as they are for today's autonomous driving. In order to deal with some of the issues we have identified in our previous work (Koester et al., 2013), we propose an entirely new approach of handling depth data for accessible section detection. Our new approach separates depth data into distinctive slices and processes them individually, aggregating the results in a later step. Furthermore, we rely on some basic assumptions, *i.e.*, the White Cane operator is standing on the accessible section, while the camera is somewhat aligned to the horizon, *i.e.*, upright. We can therefore assume that accessible section segments have their normals point upwards and aggregate segments accordingly per slice, and later over all slices, yielding the accessible section in front of the user.

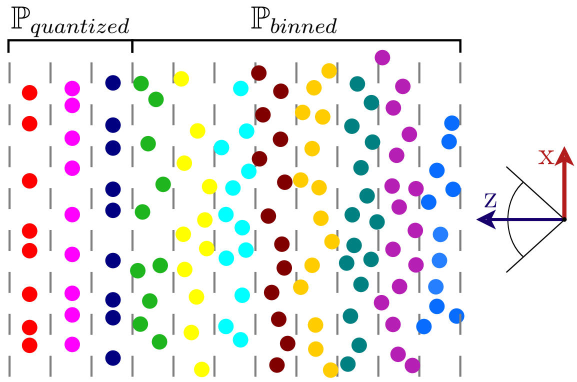


Figure 4.3: Our proposed Slice&Dice approach separates depth data into individual depth slices, based on each point's z-value, *i.e.*, its distance to the camera (colored point groups). For further processing, these can be either quantized ($\mathbb{P}_{quantized}$), mapping the points onto a plane, or left unchanged ($\mathbb{P}_{binned}$), if a binning approach is preferred to not lose any depth information, largely depending on further processing steps and required depth precision.

Slice&Dice

We propose to separate depth data into individual slices, dice them into further pieces if found necessary, process them individually, *i.e.*, parallelized, and merge their results in later processing steps. We refer to this as Slice&Dice, an idiom meaning "to divide something into many small parts especially to use the result for one's own purposes."[2] We thus separate the depth data, *i.e.*, the points of the point cloud, into thin slices of constant width, separated only by their z-values (*cf.* Figure 4.3). Although possible values for the slice width range from a few centimeters up to meters, it should be largely based on the task and the objects one intends to detect. Furthermore, each slice can be processed individually, allowing for a high parallelization. This creates a trade-off between parallelism and detectable object sizes or boundaries. For our specific use case, we have decided to use ten centimeters for the width, as it allows us to detect rather small structures, *i.e.*, accessible section in front of the user, using incremental ten centimeter wide steps, or the beginning or end of a shoreline segment with a maximum accuracy of ten centimeters as well, as considered shorelines should be perpendicular to our slices (*cf.* Section 4.2.2).

Figure 4.4 shows how our slicing approach would separate real world depth data. Please note how the noise in the depth data increases with the z-value due to anisotropic noise in the depth reconstruction, *i.e.*, small appearance-based reconstruction errors have a larger effect for far away points. At the same time, the amount

[2] https://www.merriam-webster.com/dictionary/slice and dice

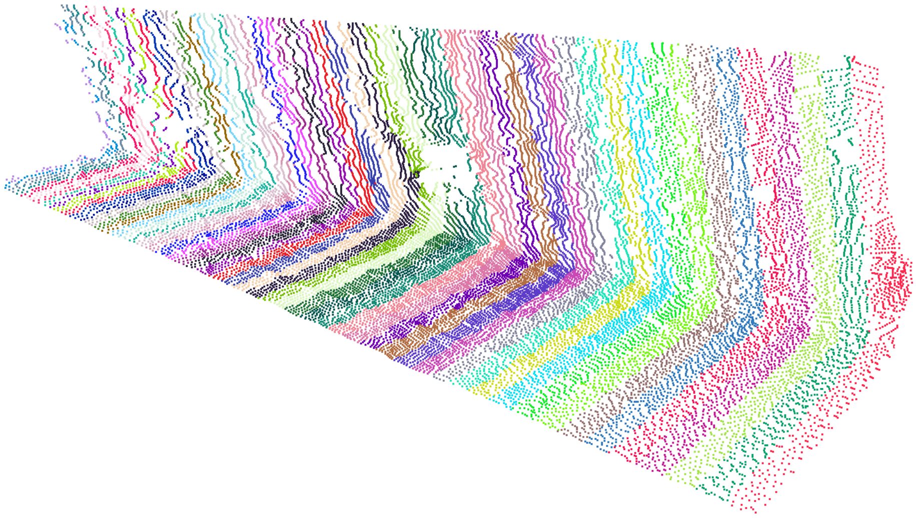


Figure 4.4: Real world data, sliced and color coded. Please note how the noise level increases significantly for very far away slices. Also, when setting all slices to a constant width, the number of points per slice decreases for further away slices due to optical constraints.

of points per slice is decreased with an increasing distance to the camera due to optical constraints, *i.e.*, further away objects map to fewer pixels on the imaging chip. While it would be possible to mitigate the second observed effect using slices of increasing width for increasing distances, we did not further investigate this idea, although it would also be a feasible possibility for our accessible section detection, as we require less precision for far greater distances compared to near-range segments.

Our slicing allows model fitting algorithms (the "dice" step), *i.e.*, RANSAC, to not only fit data to an easier model, *i.e.*, a two-dimensional line or curve *vs.* a three-dimensional plane or surface, but also considers much less data in the first place. Furthermore, this improves noise resistance, as noisy depth data gets either quantized onto a very discrete set of planes, partly reducing the noise in at least one dimension, or when using binning, the effects of noise is reduced as well, as usually a different dimension is considered in our following processing steps.

However, one has to choose the direction in which to slice very carefully, so as not to lose relevant depth information. For us, this usually means to slice along the camera's z-axis, as we are mostly interested in structures that are perpendicular to our camera's optical axis, *i.e.*, the accessible section, or objects that are somewhat parallel to it, *i.e.*, shorelines. Slicing at a large angle to this optical axis would complicate our next processing steps, as well as render our previously mentioned assumptions about the accessible section's relative position invalid.

Algorithm 2: Slice&Dice-Based Accessible Section Detection

$\mathcal{S_P} \leftarrow \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ ▷ slicing, *i.e.*, separate points by depth
for all $s_i \in \mathcal{S_P}$ **do** ▷ iterate over all slices
  $\mathcal{C} := \mathit{agglomerativeClustering}(s_i)$ ▷ Cluster points by their normals
  for all $c_j \in \mathcal{C}$ **do** ▷ iterate over all clusters
    $r_j := \mathcal{RANSAC}(c_j)$ ▷ fit line to cluster
  end for
  $\mathcal{R} \leftarrow \{r_1, \ldots, r_n\}$ ▷ store lines
  for all $r_j \in \mathcal{R}$ **do** ▷ iterate over all fitted lines (and points)
    $r_j := \mathit{shorten}(r_j, c_j)$ ▷ shorten line w.r.t. cluster points
  end for
  $s_i := \{r_1, \ldots, r_n\}$ ▷ store all segments in slice
end for
$\mathcal{S_L} \leftarrow \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$ ▷ slices with line segments
$\mathcal{A_S} := \{\varnothing\}$ ▷ initialize accessible segments
for all $s_i \in \mathcal{S_L}$ **do** ▷ iterate over all slices
  for all $l_j \in \mathcal{L}_i$ **do** ▷ iterate over all line segments
    if $\mathit{horizontal}(l_j)$ **then** ▷ consider only (almost) horizontal segments
      if $\mathcal{A_S} = \{\varnothing\}$ **then** ▷ initialize if empty
        $\mathcal{A}_i := \{l_j\}$
      else if $\mathit{height}(l_j) \precsim \mathit{height}(\mathcal{A})$ **then** ▷ if height similar or less
        $\mathcal{A_S} := \{\ldots, \{\mathcal{A}_i, l_j\}, \ldots\}$ ▷ add to accessible segments
      end if
    end if
  end for
end for
for all $\mathcal{A}_{\mathcal{S}i} \in \mathcal{A_S}$ **do** ▷ connect segments to enclosed section
  $\mathit{connectSegmentsEndsToNextSlice}(\mathcal{A}_i, \mathcal{A}_{i+1})$
end for

Accessible Section Detection

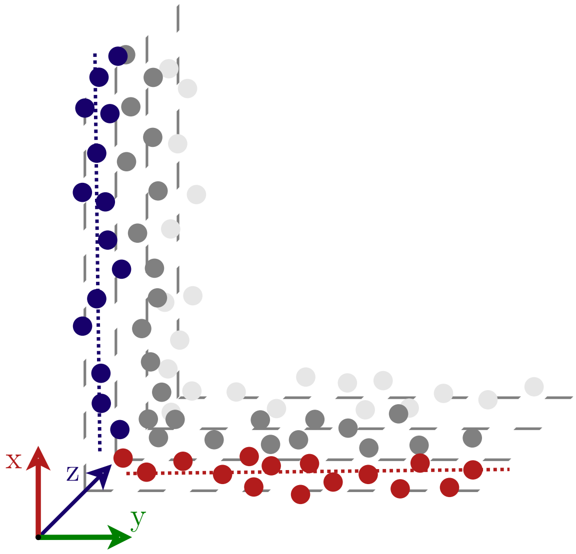
We start by processing acquired depth data with our proposed Slice&Dice approach: The data is sliced into ten centimeter wide slices for further processing, allowing a maximum accuracy of ten centimeters in the camera's z-axis. We chose this specific slice width as a trade-off between possible parallelism and detection accuracy, as a distance wise quantization into ten centimeter wide steps was found to be sufficiently precise in our prior experiments for accessible section detection, *e.g.*, noise prevents further precision for increasing depths. We use a "Stereolabs ZED"[3] camera to

[3] https://www.stereolabs.com/

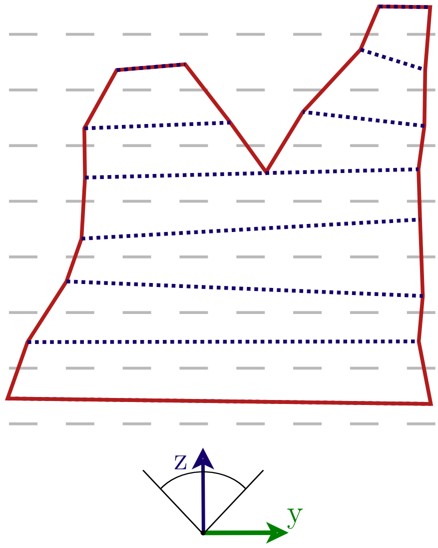
acquire the depth data, which has a vertical field of view of 60 degrees (90 degrees horizontal) and provides depth data between 0.5 and 20 meters. This provides us with ∼200 slices per frame that we have to process further, *i.e.*, cluster and agglomerate. It must be noted that due to optical constraints of any camera, *i.e.*, a limited field of view, this first slice is usually at a distance of one to two meters from the camera, assuming an average camera height of ∼1.4 meters, when carried at chest level, a vertical field of view of 60 degrees, and a horizon aligned camera — and often contains the first pieces of accessible section visible in the data. However, as the White Cane usually already considers ranges of up to almost 1.5 meters (*cf*. Section 4.1.1), any arising gap to our accessible section detection is non-existent most of the time.

Slice Wise Clustering After slicing (*cf*. Algorithm 2), the points in each slice are pre-clustered using their surface normals, which are also generated from the depth data, *i.e.*, the ZED already provides these, but it is also possible to generate them from the points in each slice using their neighbors. Line 3 uses an agglomerative clustering approach that processes each slice along the its x-axis, *i.e.*, perpendicular to the camera's optical axis, and if we assume an upright camera, from left to right. It creates a new cluster whenever it encounters a point normal that significantly deviates from the currently created clusters, *i.e.*, more than 25 degrees, as we use the *Manhattan World Assumption* (Coughlan and Yuille, 2000), *i.e.*, urban areas exhibit mostly horizontal or vertical surfaces. Line 5 then uses RANSAC to efficiently fit a line to each generated point cluster. Line 9 afterwards shortens these RANSAC matched lines to line segments, based on each clusters points (*cf*. Figure 4.5a).

Segment Agglomeration In an agglomerative approach, similar to Koester et al. (2013), we start with the first slice, *i.e.*, the one closest to the camera (*cf*. Algorithm 2). Lines 15–16 then (slice-wise) process each line segment: Line 17 performs a *horizontal*(...) check for each line, which in two dimensions only has to consider the slope of the line, again assuming a somewhat upright camera. Lines 18–22 then collect all horizontal line segments: We agglomerate only the *lowest* one — or multiple ones if they share the same *relative height*. We then proceed to the next slice, where we again search for horizontal line segments that match the *relative height* of our already collected segments. Lines 26–28 create our final detection from the segments in $\mathcal{A}$, *i.e.*, an actual enclosed section and not just individual line segments, by connecting the ends of each segment to the nearest segment ends of an overlapping next slice, if there are any available (*cf*. Figure 4.5b).



(a)



(b)

Figure 4.5: We cluster each slice's points and use RANSAC to detect line segments (a). These are then pruned using their slope, only keeping horizontal segments. Our final agglomeration step, starting with the closest slice, iteratively collects these line segments and connects them into an enclosed section, *i.e.*, the final accessible section detection (b).

4.1.3 Evaluation

Although we recorded a dataset using the ZED camera (*cf*. Section 4.2.3), we cannot evaluate on this dataset, as there are no accessible section labels available for it and manually creating these is extremely time consuming, as we know from prior work. Table 4.1 therefore compares our Slice&Dice approach to our previously proposed algorithm that is also based on depth data (Koester et al., 2013), *i.e.*, on the FlowerBox[4] dataset we had created for its evaluation. Interestingly, our proposed algorithm, based on the Slice&Dice technique discussed previously, achieves very similar results, it performs only slightly better on average. This is mostly due to it using a very similar approach to gather the accessible section in its final processing step, *i.e.*, it agglomerates horizontal plane-stripes between the line segments of two neighboring slices, whereas in Koester et al. (2013) we collected ground plane patches whose plane normal was pointed upright for a camera aligned to the horizon, but also did this starting on the lower image border, iteratively considering the next patch from bottom to top, for each vertical column of ground plane patches. Therefore, it

[4] https://cvhci.anthropomatik.kit.edu/people_dkoester.php#FlowerBox

Table 4.1: Accessible section detection results compared to Koester et al. (2013) on the FlowerBox dataset. We compare: *Area under the curve* (AUC) for $\int\mathcal{ROC}$ and *precision vs. recall* ($\int\mathcal{PR}$), as well $\mathcal{F}_\beta$-scores for $\beta = 0.5$ and $\beta = 1$, and finally the accuracy ($\mathcal{ACC}$).

	Koester et al.					Slice&Dice				
	$\int\mathcal{ROC}$	$\int\mathcal{PR}$	$\mathcal{F}_{0.5}$	$\mathcal{F}_1$	$\mathcal{ACC}$	$\int\mathcal{ROC}$	$\int\mathcal{PR}$	$\mathcal{F}_{0.5}$	$\mathcal{F}_1$	$\mathcal{ACC}$
Alley	.928	.882	.937	.916	.901	.947	.901	.949	.922	.913
Alley L.	.892	.856	.941	.911	.862	.935	.896	.959	.924	.894
Bicycle	.753	.629	.843	.869	.676	.780	.653	.857	.874	.685
Car	.850	.679	.763	.739	.851	.867	.690	.774	.745	.871
Corridor	.819	.665	.816	.750	.796	.866	.674	.822	.764	.804
Fence	.855	.750	.878	.834	.815	.871	.772	.885	.848	.832
Flower-box	.783	.607	.838	.789	.724	.838	.651	.869	.813	.783
Hedge	.836	.827	.882	.872	.814	.862	.853	.899	.880	.828
Ladder	.836	.629	.757	.736	.868	.798	.682	.814	.802	.823
Narrow	.958	.924	.922	.928	.929	.969	.941	.947	.936	.953
Pan	.759	.548	.843	.861	.650	.788	.632	.883	.870	.677
Passage	.850	.733	.889	.821	.805	.863	.745	.897	.827	.824
Railing	.760	.626	.842	.852	.696	.728	.606	.851	.855	.650
Ramp	.803	.680	.870	.839	.731	.815	.701	.889	.855	.755
Ridge	.854	.622	.230	.304	.199	.837	.603	.209	.270	.277
Sidewalk	.929	.945	.943	.947	.913	.945	.956	.958	.962	.937
Sidewalk 2	.947	.914	.913	.912	.904	.969	.938	.927	.919	.913
Sidewalk L.	.889	.942	.954	.950	.912	.913	.968	.971	.964	.925
Sign	.890	.835	.933	.899	.854	.922	.849	.945	.919	.869
Street	.940	.885	.919	.904	.917	.964	.912	.934	.925	.926
∅	.857	.759	.846	.832	.791	**.874**	**.781**	**.862**	**.844**	**.807**

also suffers from similar inaccuracies when assembling the individual parts in the entire accessible section and loses a lot of precision in the border regions.

While the area under the ROC curve ($\int\mathcal{ROC}$) as well as the precision *vs.* recall curve ($\int\mathcal{PR}$) only improves by a few percent, the accuracy ($\mathcal{ACC}$) is almost unchanged. However, the differences between the two $\mathcal{F}$-scores are more interesting to us, *i.e.*, the improvement is slightly larger for $\mathcal{F}_{0.5}$ (+1.6%) when compared to $\mathcal{F}_1$ (+1.2%), at least suggesting that while still an insignificant change in both cases, our current approach detects the accessible section at a slightly higher precision, which we prefer over having a higher recall, since we consider this a safety critical feature.

4.1.4 Conclusion

We propose an approach for accessible section detection based on depth data, that implicitly also considers its dual problem — obstacle detection. Therefore we introduce our Slice&Dice technique, which allows us to separate depth data into individual slices and further process these separately. This has the added benefit of allowing us to highly parallelize this computation and also simplify some computational steps, *i.e.*, we can now focus on the detection of horizontal point clusters within two dimensional planes, instead of having to detect a plane in three-dimensional data.

For our evaluation, we compare our current approach to a previously proposed technique (Koester et al., 2013) on an accessible section detection dataset, *i.e.*, FlowerBox, and achieve slightly better results. The evaluation also suggests a slightly higher precision, which fits with our previously defined key concepts (*cf.* Section 2.5), as detecting the accessible section in front of a user is a safety critical feature, and therefore precision should be valued higher than recall in such a scenario. If we detect accessible section in front of the user, it's better to err on the side of caution, *i.e.*, we would rather miss a part of accessible section than to provide a false positive detection, potentially causing harm to the user that trusts and relies on our information. Consequently, we must further investigate this approach's issues, in order to improve the overall accuracy of our algorithms, not just with respect to precision — for recall as well — and create a highly reliable accessible section detection system in the near future, as part of a complete Orientation&Mobility system. Figure 4.2 shows an example of our generated accessible section on a pedestrian walkway, where the accessible section extends forwards between a parked car on the left and a building's facade on the right.

It must be noted that, although accessible section detection techniques increase the perceivable range of the White Cane, it is not straightforward to communicate this information to the cane operator. This is a field that also requires further research, as providing this information to the operator must be performed in a way that is not overwhelming to the user, but still increases spatial awareness for the immediately available accessible section. Ideally, we would integrate this technique with other detection methods, such as the shoreline detection technique presented in section 4.2 and also integrate high-level information from O&M's orientation aspect.

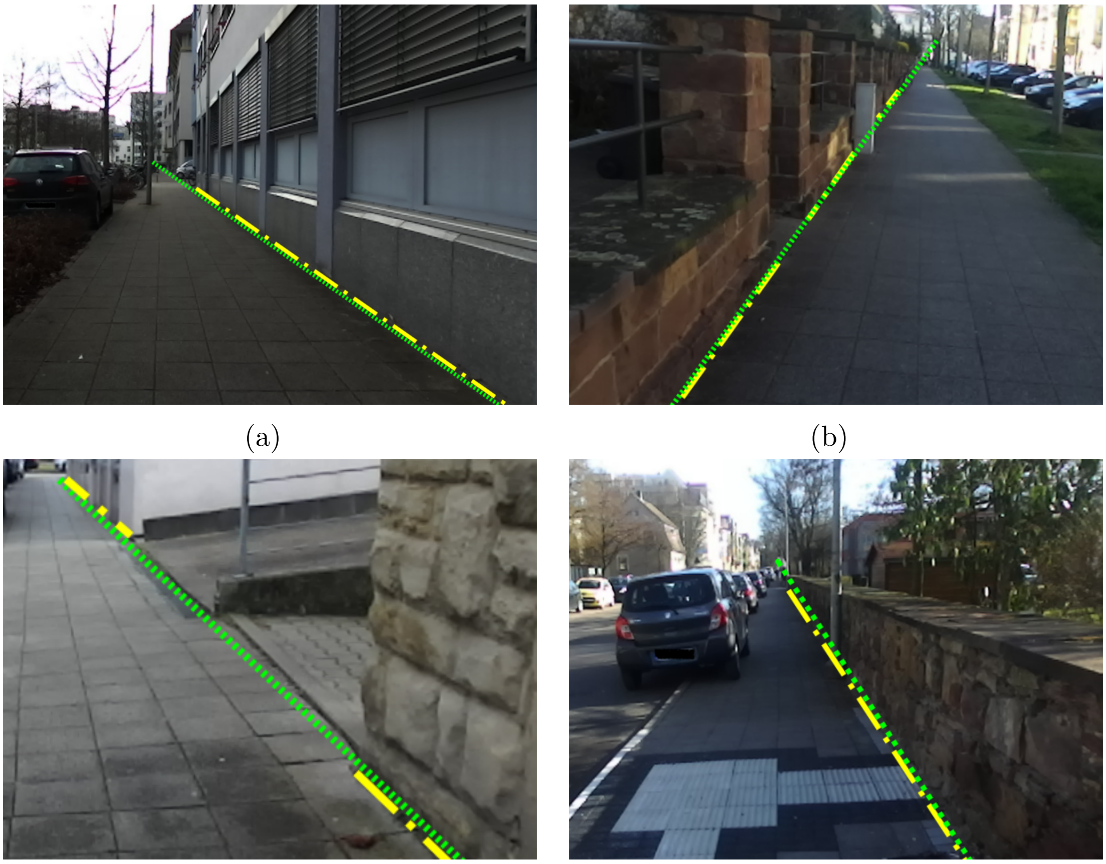
4.2 Shorelines

Shorelines are an important aspect of Orientation & Mobility White Cane techniques, as they provide people with visual impairments with a physical object to follow along for a prolonged period of time. O&M roughly differentiates *inner* shorelines — the junction of accessible section and walls — and *outer* shorelines — the junction of pavement and curblines — in a pedestrian walkway scenario. However, generally speaking, each haptic edge can become a *natural* shoreline, as long as it is sufficiently perceivable with the White Cane, *e.g.*, walkway borders towards grass or a small path of cobble stones. In addition to *natural* shorelines, there are also some artificial ones being used, *e.g.*, tactile paving that is installed in ever more locations like on train platforms, tram stations, or around zebra crossings (*cf*. Figure 3.3).

An inherent problem between shorelines and the White Cane is that shorelines are imperceptible from a distance. We therefore propose to help locate shoreline from afar and notify about their existence, so that we can provide guidance for people with visual impairments in locating those in the first place. Such an enhancement of the White Cane is especially useful after pedestrian crossings, passing driveways, going around corners, and many other situations — every time a shoreline exhibits a discontinuity due to its physical nature. This is particularly beneficial in unknown locations, as it is almost impossible for people with visual impairments to already have a decent spatial awareness of their surroundings in these situations.

4.2.1 The Shorelining Technique

Shorelining is an essential part of O&M. Therefore, we investigate its usage and try to provide additional assistance, enhancing the White Cane's abilities with computer vision, as this important O&M technique is used for indoor, as well as outdoor scenarios. While shorelining requires the operator to swing the cane in an arcing motion in front of the body and to touch the ground shortly on the opposite side of the forward facing foot, there also exist other techniques such as trailing, where the cane is merely held as a protection while the back of the hand is used to follow along a wall — our proposed approach can provide assistance in both cases. The cane should always be on the opposite side of the forward foot in order to protect, and constantly check the accessible section in front of, the cane operator, *before* the



(a) (b) (c) (d)

Figure 4.6: Four natural shoreline examples in urban areas, mostly building facades and fences. The green dotted line is our detected shoreline location, while the yellow dashed line represents the reference label: (a) a building's facade, (b) a stone-fence, (c) a driveway creates a gap in the shoreline, and (d) another stone fence, where (b-d) are zoomed in cutouts for improved visibility. In all four cases our algorithm very accurately locates the shoreline's position and direction, even when it has large gaps or continues further away.

backward foot is going to move forward. A natural shoreline is merely touched from time to time in this process, while the cane operator follows along its direction.

We therefore detect, locate, and track inner shorelines, to inform the user about their existence, relative location, endings, and possible discontinuities beforehand. We also provide a straight path between two natural shoreline segments. This process increases spatial awareness and allows to more comfortably explore terrain self-sufficiently. Since this assistive approach was partly inspired by our prior work on shoreline level routing (*cf*. Section 3.2), we reuse the *real* and *virtual* shoreline definitions from section 3.2.2 (*cf*. Figure 3.12). Relevant related work to this approach is discussed in section 2.3.2, although it has to be noted that, to the best of our knowledge, so far there exist no other work that tries to assist in shoreline related issues from a body worn camera. Only Coughlan and Shen (2007); Coughlan et al. (2006) and Ivanchenko et al. (2008a) proposed curb and shoreline detection systems, however, these were intended for blind and partially sighted wheelchair users, a specific application which provides a very stable camera platform and also mostly detect drop-offs, but do not track shorelines and assist in maneuvering their gaps.

4.2.2 Shoreline Localization

We propose a novel shoreline detection approach that tries to visually detect and track shoreline segments from a first person view, *i.e.*, a body worn camera that provides depth data to us. We focus on a mostly urban scenario, as our focus lies on inner shorelines, *i.e.*, the junction between the pedestrian walkway and building facades or walls. Gaps or discontinuities in shorelines are also considered, as we track individual segments and virtually fill these (*cf*. Figure 4.6). We furthermore can inform about reachable shorelines in unknown locations. Ideally, this kind of information can benefit from, as well as integrate itself into, routing approaches, such as our impairment aware routing presented in section 3.2.1.

Our shoreline detection and localization algorithm is based on the *Manhattan World Assumption* — where cities that are build on a cartesian grid lead to prominent edges in taken images (Coughlan and Yuille, 2000). This lets us assume that in urban areas a White Cane operator will often follow along a natural shoreline, which most likely points at a vanishing point. We localize this vanishing point using image-based approaches and track its position in real world coordinates over time. This allows us to determine a ROI for our shoreline detection, as shorelines are normally located below the vanishing point to the left or right side, which could be retrieved from shoreline level routing information. We also track shorelines and their individual segments over multiple image frames, which greatly stabilizes the detection over time, as can be seen in the evaluation. Our only other assumption is the existence of a vertical segment adjacent to the ground plane, which is mostly true for urban areas, but we also consider not completely flat or absolutely vertically surfaces, *i.e.*, hedges or even cars parking in a row.

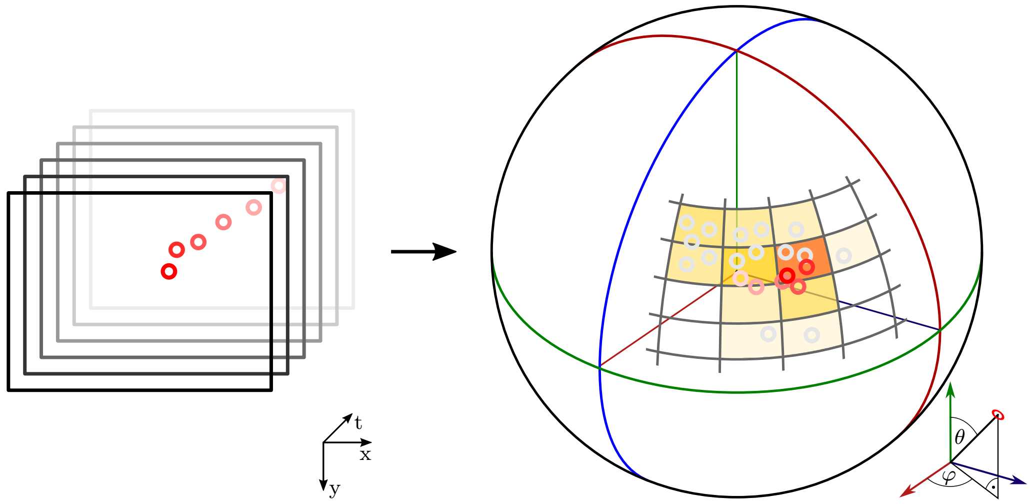


Figure 4.7: We project the frame wise detected vanishing points from image coordinates into spherical coordinates, add these into an accumulator array, and use a frame wise decay to finally report the cell with the maximum value as our tracked vanishing point over time.

Visual Odometry. In this work, we rely on the "Stereolabs ZED" camera that provides depth data and visual odometry as well. When we cannot use the ZED or don't have an odometry available, we also rely on "LIBVISO2"[5] (Geiger et al., 2011). With a visual odometry, we can simply estimate our general direction of movement by averaging the location differences of our last ten camera frames — the user's last locations. We rely on this estimated direction to detect and score possible shoreline candidates located in front of the White Cane operator, since a shoreline that is being followed should be aligned with our walking direction, or not be relevant to us. Furthermore, in a Manhattan World, the shoreline should also be aligned with the most dominant vanishing point. If a detected shoreline meets both these assumptions, it significantly increases its likelihood of being a relevant shoreline candidate to us.

Vanishing Points. We use an image-based — not depth-based — technique to detect vanishing points in each frame. Our approach is similar to Wu et al. (2016), where one first difference is that we use another line detection approach, proposed by Akinlar and Topal (2011). Their EDLines system is a very fast straight line search algorithm. We use these line segments and immediately discard any almost horizontal or vertical segments, as they can't point towards our vanishing point — in a Manhattan World only diagonal and similar edges point to the vanishing point we are interested in. We then weight our remaining line segments based on their diagonality, *i.e.*, they should

[5] http://www.cvlibs.net/software/libviso/

be at an angle close to 45°, and length, *i.e.*, longer segments are weighted higher as well:

$$\omega_i = \frac{|\mathcal{S}_i|}{\mathcal{I}_{diag}} * exp\left(\frac{(abs(\theta_{\mathcal{S}_i} - 90) - 45)^2}{-2 \cdot (45)^2}\right) \quad , \tag{4.1}$$

where $\mathcal{S}_i$ is the current line segment, $\mathcal{I}_{diag}$ the maximum line length (image diagonal) to normalize the segments lengths, and $\theta_{\mathcal{S}_i}$ the segments angle (between 0 and 90), where the angular weight is normalized as well. Finally, we formalize a RANSAC model (Fischler and Bolles, 1981) to determine the vanishing point candidate-based on our remaining segments. This point is the projection from the image into real world coordinates and tracked over multiple frames. The vanishing point detection could also be performed in depth data, but we did not investigate it further. The frame-based vanishing point is tracked over time using spherical coordinates, implemented as a two-dimensional accumulator array with built-in decay, *i.e.*, we represent the spherical surface in an array, where the maximum value over all cells yields our averaged vanishing point coordinates (*cf*. Figure 4.7). This process stabilizes the vanishing point over time in our real world coordinate system, complicated by our constantly moving camera. Thus, when we maneuver around a corner or perform a turn in general, the old vanishing point eventually disappears, as it simply decays over time, while a new candidate will be acquired over time in the same manner.

Shoreline Detection

After having calculated the vanishing point in real world coordinates, we can use it to score shoreline candidates. We detect these candidates in each individual frame, but also agglomerate them over time, always in real world coordinates. Finally, we always only report the shoreline with the highest confidence level in this scenario.

We base our shoreline detection on the same Slice&Dice approach as described in section 4.1.2 (*cf*. Figure 4.8). The ZED stereo camera not only provides depth data as a point cloud, but also calculates surface normals per point. We use these surface normals to distinguish points between ground (an ideally upright normal) and walls (an ideally horizontal normal) and cluster accordingly, *i.e.*, we first collect all normals in an accumulator array (*cf*. Figure 4.7), and then identify the two largest clusters that are ideally separated by exactly 90 degrees (horizontal ground plane *vs*. vertical building facade). This approach allows us to not require a horizon aligned camera

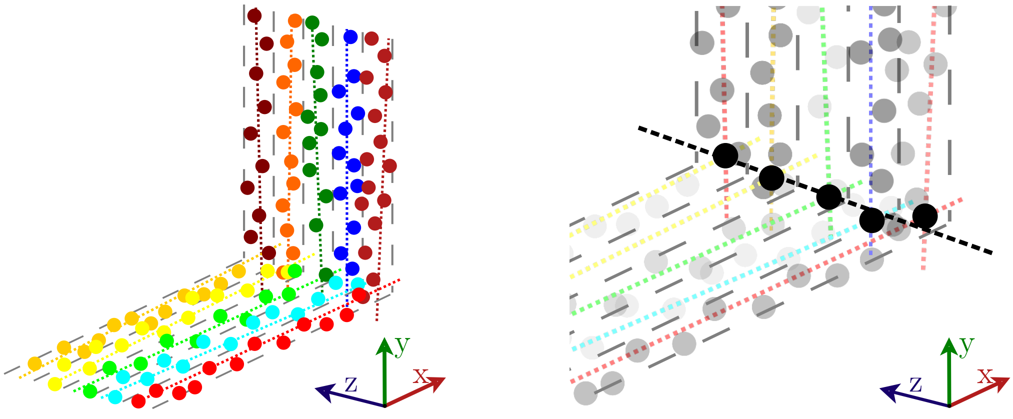


Figure 4.8: Our Slice&Dice-based clustering (*cf*. Section 4.1.2). We cluster points slice-wise according to their surface normals, use a weighted RANSAC to detect almost horizontal as well as sufficiently differing line segments, and determine their 2D-intersections in the slice's main plane (a). We then generate a shoreline candidate from these intersections (b).

and is thus much robuster to inadvertent camera rotations. Finally, we again rely on RANSAC, however here we use "Optimal Randomized RANSAC" by Chum and Matas (2008), and an implementation provided by Raguram et al. (2013), where our weight is the normalized angular difference for each point's surface normal and an assumed ideal normal, *i.e.* the one we identified for our two largest clusters.

This provides us with one line per cluster and by calculating their intersections, where we assume all slice's points to be within a plane, we gather shoreline segment candidates per slice, which we compute in a highly parallel manner. The actual shoreline candidate itself is then determined, again based on a weighted RANSAC, which tries to fit all shoreline segments onto a single line in our real world coordinates.

Shoreline (Segment) Tracking

We use slices of roughly ten centimeters in depth, thus our shoreline segment length is ten centimeters as well. We track these frame wise generated segments in real world coordinates over time, as well as the whole shoreline itself, to increase robustness.

We accumulate individual shoreline hypotheses, constantly merging sufficiently close ones, where their score is added as well, similar to the accumulator array used to track vanishing points. We only receive a single shoreline candidate per frame, this step is therefore not computationally expensive, as we usually only have to track a few hypotheses over time, often merged into a single hypotheses after a few more frames, due to the robustness to noise of the entire detection pipeline. We also use a decay over time, similar to the vanishing point detection, to slowly fade out old hypotheses and report only the highest scoring hypothesis of our hypotheses pool.

To track shoreline segments, we use the same approach as for shoreline hypotheses, however, here we do so per slice. This requires the additional step of having to match slice-based segment hypotheses in our current frame to the ones we already have. We do this by simply calculating the slice offset in real world coordinates, using the next best match. In this process, similar to whole shoreline hypotheses tracking, we create many of such hypotheses for each shoreline segment and also for each real-world coordinate slice. While this might seem to be much more expensive processing-wise, it is actually not so, since after the initial matching is done, each slice is updated independently of the other slices, allowing for a parallelized process here as well.

Shoreline Discontinuities. We furthermore deal with the issue of shoreline discontinuities, or gaps. Luckily, our shoreline segment hypotheses tracking already contains this information. When we cannot find a segment for a particular real world location, a gap develops over time, as no hypotheses can be generated for that real world slice. This could either mean that a shoreline ends or that there is a discontinuity. We can now use our tracked shoreline candidate, project it through all the segment hypotheses of each slice, and check whether there exists a segment in each slice with a sufficiently high segment score. That way we can find out, whether the shoreline has ended or just exhibits a gap, which means we can virtually close the gap and assume the shoreline continues in the future. Again, very precise shoreline level routing information would be very helpful in these cases.

4.2.3 Evaluation

We evaluate our proposed shoreline detection and tracking algorithm by recording sixteen sequences in an urban area (*cf*. Figure 4.6), along multiple street sections with building facades and gaps in-between, where available shoreline segments where labeled afterwards. We analyze the three-dimensional angular deviation, *i.e.*, the angle between two vectors from the same origin, here ignoring the distance offset. Furthermore, we report the minimal distance offset, *i.e.*, the three-dimensional averaged distance of a proposed shoreline candidate to each point on our labeled

Table 4.2: Results for our Slice&Dice-based shoreline tracking approach, comparing frame wise and tracked detections, where angular errors are given in degrees as mean ($\bar{\theta}$), median ($\tilde{\theta}$), and standard deviation (δ_θ), and similarly for distance errors in centimeters ($\bar{d}$, $\tilde{d}$, δ_d).

	frame wise						tracked					
	$\bar{\theta}$	$\tilde{\theta}$	δ_θ	$\bar{d}$	$\tilde{d}$	δ_d	$\bar{\theta}$	$\tilde{\theta}$	δ_θ	$\bar{d}$	$\tilde{d}$	δ_d
1	1.16	0.86	1.37	2.66	2.41	2.14	0.87	0.71	0.84	2.29	1.57	2.01
2	1.49	0.91	1.85	5.39	2.90	14.69	1.34	0.74	1.76	9.33	2.06	27.41
3	1.62	1.12	1.57	5.26	3.92	9.76	1.08	0.70	1.06	4.23	3.74	3.76
4	1.18	0.85	1.23	4.09	3.63	3.38	1.14	0.65	1.89	3.84	2.67	5.62
5	1.23	0.71	3.50	11.06	4.20	56.90	1.15	0.52	2.19	6.37	4.20	32.94
6	1.61	0.90	2.60	7.35	2.63	22.57	1.42	1.06	1.21	3.30	2.59	9.70
7	1.48	0.97	4.01	3.62	2.33	5.99	1.24	0.90	1.79	2.70	1.96	2.55
8	1.42	0.85	3.00	16.21	2.87	125.79	1.18	0.79	1.06	3.87	2.69	10.57
9	1.58	0.95	2.84	8.12	3.51	17.00	1.30	0.96	1.73	4.33	1.97	12.54
10	1.60	1.06	1.50	4.51	2.47	8.26	1.57	1.02	1.62	2.78	1.93	7.74
11	1.74	0.98	2.37	3.56	2.45	4.05	1.23	0.78	1.55	4.68	2.36	6.05
12	1.18	0.85	1.06	3.46	2.81	2.82	1.15	0.83	1.06	3.92	2.86	3.16
13	1.85	0.97	6.23	3.82	2.30	15.60	1.16	0.76	4.17	5.01	2.49	6.57
14	1.76	1.03	5.40	3.60	2.74	10.18	1.29	0.97	1.06	2.80	2.20	2.74
15	1.36	0.95	1.95	11.72	3.20	38.62	1.26	0.88	1.67	4.16	2.60	15.70
16	1.46	0.99	1.50	4.44	3.26	5.90	1.19	0.74	1.36	6.20	2.32	22.70
∅	1.48	0.93	2.62	6.18	2.98	21.48	1.22	0.81	1.63	4.36	2.51	10.74
Δ							-18%	-13%	-38%	-29%	-16%	-50%

groundtruth segment, as shoreline candidate and label are usually a set of skewed lines that do not intersect each other. Table 4.2 shows that our proposed hypotheses tracking and merging algorithms significantly reduce angular and distance errors (compared to frame wise). While the distance error's standard deviation is halved, mean and median errors are reduced up to almost one third. We also evaluate our shoreline detection algorithm on a per segment basis. Here, the slice depth for the Slice&Dice approach is 10 centimeters, which also sets the individual shoreline segment length to be 10 centimeters. We choose this specific size, as it represents a good compromise between speed and spatial resolution, where for example half a meter would have been way to coarse for our intended fine-grained guidance assistance. Furthermore, it would be very labour intensive and error prone to create labels with a much higher precision, *i.e.*, less than a few centimeters. Figure 4.9 shows that, although depth data usually degrades with increasing distance due to a significant

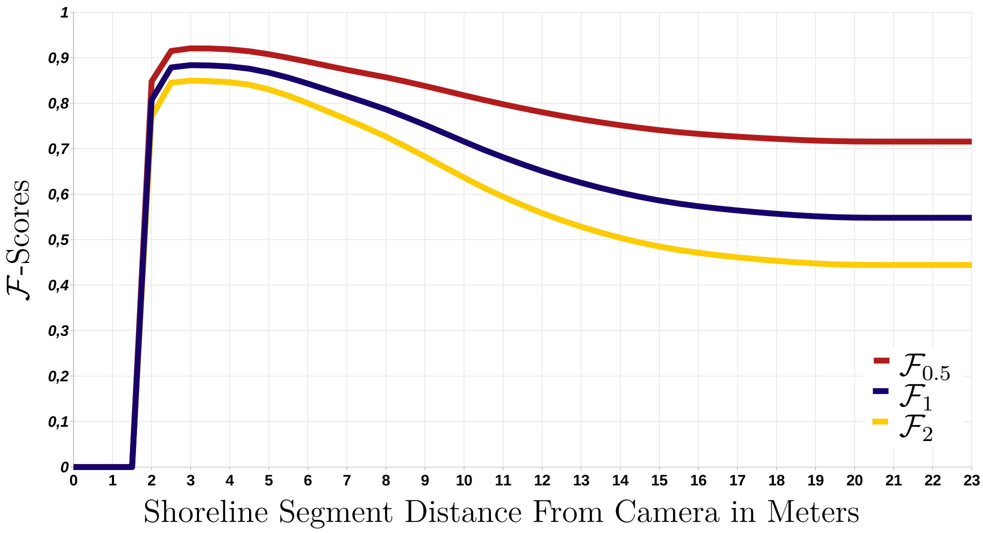


Figure 4.9: Shoreline segment-based evaluation, where we compare individual segment $\mathcal{F}_\beta$-scores for all encountered segment distances from the camera up to 23 meters.

increase in noise levels, our $\mathcal{F}$-scores drop, between three and twenty meters, by $\sim$0.4 for $\beta = 2$, by $\sim$0.35 for $\beta = 1$, and finally, only by $\sim$0.2 for $\beta = 0.5$. We thus conclude that if we detect a segment, we do so with a very high confidence, as the comparison between the different $\mathcal{F}$-curves shows. This is an especially important factor of our system, as we are again, similar to the accessible section detection, providing safety critical information to the user that should be as reliable as possible.

4.2.4 Conclusion

We proposed a novel algorithm to detect shorelines from a first person viewpoint, *i.e.*, from a White Cane operator's perspective. Our approach relies on the detection of vanishing points and visual odometry and detects and tracks natural shorelines, in segments of varying size. It shows very promising results, as it detects shorelines very accurately and reliably. Furthermore, the included segment-based information allows our approach to provide information to the operator when a shoreline ends or to bridge discontinuities, *i.e.*, to guide the operator towards the next shoreline. It would be very beneficial to integrate this approach with our shoreline level routing system proposed in section 3.2.1, as a combination of both systems could provide great assistance in shorelining. Moreover, an adaptation to less visible natural shorelines is needed, *i.e.*, curblines or low border stones. Finally, the actual usefulness of our proposed approach should be further investigated in a follow-up user study.

4.3 Pedestrian Crossings

Pedestrian crossings provide an important aspect in urban areas — a means to safely cross the street, even in dense traffic. However, for people with visual impairments, available crossings are often a major concern and cause of insecurity (Matthews et al., 2014). While specific crossing installations, such as pedestrian traffic lights at intersections and zebra crossings, usually located a roundabouts or in the middle of a street, are commonplace, these provide little to no accessibility features by default.

4.3.1 Crossing Accessibility

Zebra crossings are rarely encompassed by tactile paving on the sidewalk, so people with visual impairments might just as well walk straight by — without taking notice of their existence. Pedestrian traffic lights, while much safer to use — when they are accessible — present multiple challenges. First, the correct signal request pushbutton has to be located and pressed. Ideally, it would contain relief symbols that describe not only the crossing direction, but also information about the number of lanes and other obstacles. Then, the current walk light state must be known, *i.e.*, whether it is time to start the crossing process or continue to wait for the walk signal.

While there are visual markers, *e.g.*, thin white lines perpendicular to car traffic, that separate the pedestrian crossing surface from the general road surface, these are also hardly perceptible with a White Cane. This often causes people with visual impairments to veer off outside this specific area, especially on busy intersections, where lots of people and cyclists are all crossing at once and might bump into each other. Also, there might be additional noise from nearby cars in other traffic lanes, and they might walk straight into waiting cars, leading to further disorientation and, worst case, injuries (*cf*. Figure 4.10a). Finally, common issues are also crossing the street too early, too late, or simply too slow before traffic starts again.

Currently, only few accessibility features are widely deployed, although recently legislation in some countries is forcing communities to include APSs when creating new infrastructure. However, different regulations might contradict each other, *e.g.*, lowered curblines for people with physical disabilities, that allow them to cross a street with a wheelchair, conflicts with the interests of people with visual impairments,

(a) (b) [6] (Eliseo Velázquez Rivero)

Figure 4.10: Common accessibility issues that also affect sighted persons: (a) a car invades a zebra crossing where a pedestrian might possibly walk into it, (b) a very confusing and complex traffic light for cars, bicycles and buses, as often found in Germany.

that rely on the same curbline to orient themselves before the crossing. While O&M training tries to compensate such issues for often traveled areas, *e.g.*, from home to work, it cannot provide these benefits when maneuvering unfamiliar terrain. Furthermore, intersection layouts can be very confusing, *e.g.*, throughabouts or complicated Y-junctions with multiple traffic lanes and participants (*cf*. Figure 4.10b).

We propose to improve this situation with an assistive system that provides low-level, turn-by-turn style, guidance information, similar to shoreline-based routing (*cf*. Section 3.2), but for pedestrian crossings. More precisely, before even reaching the crossing — given we know from routing data that the user wishes to cross here — we inform the user about direction and distance of a signal request pushbutton. Next, we not only communicate the traffic light's walking state in real time, but also the distance and direction of the traffic light before and during the crossing process, where in combination with the crossing area's borders, we can prevent the user from veering of into traffic, constantly providing updates about the crossing direction and distance until the pedestrian walkway is reached. We also provide similar information for zebra crossings, *i.e.*, location, direction and distance. Finally, we seamlessly integrate our low-level pedestrian crossing guidance into our high-level routing information. This information improves spatial awareness before and during the crossing and assists in general crossing layout understanding. We present related work relevant to pedestrian crossings in section 2.3.3 and discuss a user study in section 5.2.

[6] https://commons.wikimedia.org/wiki/File:Invading_pedestrian_crossing.jpg
CC BY-SA 4.0 https://creativecommons.org/licenses/by-sa/4.0/

4.3.2 Pedestrian Traffic Light & (Zebra) Crossing Detection

Recently, deep learning methods have completely revolutionized computer vision (*cf*. Section 1.2). While, before this paradigm switch, features were created in an involved manual hand-engineered process, deep learning allows to train these, as well as the classifier, fully automatically (Krizhevsky et al., 2012). Furthermore, it achieves better accuracy in almost all domains it has been tried so far and allows us to tackle problems that were very hard before, however it always requires a sufficient amount of annotated training data. Especially important in our case is the fact that a combined object detector, *i.e.*, a detector that tries to detect and locate multiple classes, has a significant advantage over a system of detectors that are separated from each other. The successful detection of any given object class can provide further hints for other object locations, as especially in our case, the relative physical location between relevant objects, *e.g.*, pedestrian traffic lights and the poles they are mounted on, are rather fixed.

Although deep learning promises great advantages, we also have to carefully evaluate its usage. While a hand-engineered feature or classifier might require more manual labor, it also allows us to explicitly reason about its expected boundaries, failure cases and general working conditions. Deep learning does not yet provide the same level of reasoning about such issues, however much work is currently being done to achieve a better understanding and reasoning about the created networks and their limitations.

Implementation Details. Nonetheless, we decided to use a deep learning approach to detect pedestrian crossings. We use different versions of the widely popular You Only Look Once (YOLO) framework (Redmon and Farhadi, 2017; Redmon et al., 2016). YOLO is based on the "Darknet" (Redmon, 2013–2016) deep learning library written in the rather low-level programming language $\mathcal{C}$. We decided to use YOLO as its implementation details allow us to easily integrate it into our already existing software framework. Furthermore, YOLO is a quite fast object detector network, compared to other available options (*cf*. Figure 4.11), especially important to later create a real time capable prototype to be used in further user studies, therefore we based our system on YOLOv3 (Redmon and Farhadi, 2018). After evaluating different hyper parameters, we settled on a grid size of 13x13, with 5 anchor boxes each. This setup provided us with the best performance for our specific use case,

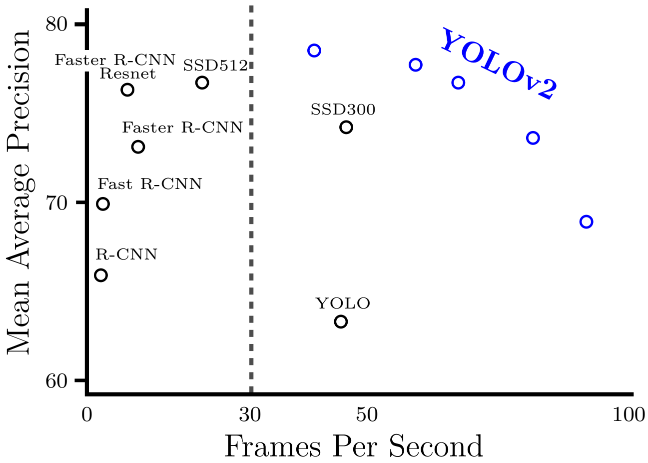


Figure 4.11: YOLO's speed *vs.* detection precision on the VOC challenge. YOLOv2 provides a great speedup when compared to other available deep learning architectures and even YOLOv1. © Redmon and Farhadi

regarding the relevant number of classes we want to detect as well as their averaged bounding box aspects. We rely on a pre-trained network provided by the YOLO authors, where the "COCO" (Lin et al., 2014) and "PASCAL VOC" (Everingham et al., 2010) datasets were used. We then refine the network on our own data, where we keep the pre-trained feature layers, and only re-train the last eight, *i.e.*, the final convolution layers and the detection layer, of the 107 layers that YOLOv3 uses.

Data Requirements. In addition to pre-training on COCO and PASCAL VOC, which YOLO already supports training on, we have created our own dataset. This was necessary, as no existing dataset contained all the labels we require to truly assist in pedestrian crossings. While there exists a multitude of autonomous driving related datasets today, *e.g.*, "KITTI"[7] (Geiger et al., 2013) or "Cityscapes"[8] (Cordts et al., 2016), these were found infeasible due to the greatly differing use case. Relevant objects are perceived from a completely different angle, *i.e.*, the road, not from a pedestrian walkway. Furthermore, marking strips, pedestrian signals or signal request pushbuttons are not contained within the labeled objects. Finally, the camera in these kinds of dataset usually moves very stable and predictable, being fixed to a car, which allows for nice tracking and reasoning, while for our use case neither is possible.

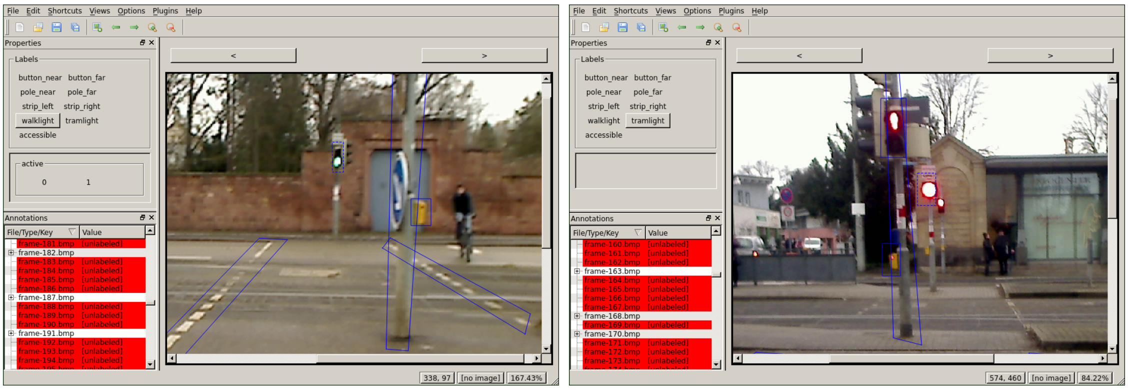
[7] http://www.cvlibs.net/datasets/kitti
[8] https://www.cityscapes-dataset.com/

Table 4.3: Details for our pedestrian detection dataset, used to train and evaluate our YOLO-based networks. Further details about the classes are provided in the text.

	$\sum$	training	validation	testing
zebracrossing	1406	784	531	91
zebrasign	1559	1203	232	124
bluesign	213	173	6	34
button	3734	2891	206	637
pole	6820	5130	627	1063
strip_left	4890	3556	456	878
strip_right	4781	3512	397	872
crosswalk	5461	3985	512	964
tramlight	1012	742	7	263
walklight	3114	2333	170	611
walklight_active	1459	1099	188	172

We therefore recorded 94 videos and randomly split these the following way: 69 videos for training, 15 videos for validation, and 10 videos for testing. Our videos include roughly 1400 instances of labeled zebra crossings (frame wise), 1600 zebra crossing signs, almost 4000 pushbuttons, 10000 crossing boundary stripes (left and right markers), and 4800 pedestrian traffic lights including their state, as well as labels for some other classes not immediately relevant for this use case, more details about the individual classes and their precise split can be found in table 4.3. The class names are provided as they are used internally in YOLO for training and evaluation, *i.e.*, "zebracrossing" refers to zebra crossings, "zebrasign" refers to blue traffic signs that indicate an upcoming zebra crossing to drivers, "bluesign" are other signs in the data to prevent confusions, *i.e.*, with street name signs, "button" refers to the yellow signal request pushbuttons, "pole" is the pole that a traffic light is mounted onto, "strip_left" and "strip_right" are the crossing area pavement markings, while "crosswalk" is their combination, "tramlight" is a specific type of pedestrian walk light that is only active to warn of an oncoming tram, and finally "walklight" and "walklight_active" refer to red and green pedestrian walk lights. Since we also use common data augmentation techniques, *i.e.*, mirroring (horizontal only), adding translational jitter, slight rotations, minor shearing, and scaling along a single image axis, these numbers grow significantly. An example of our data and its labels, generated by our in-house labeling GUI "sloth"[9] are presented in figure 4.12.

[9] https://github.com/cvhciKIT/sloth

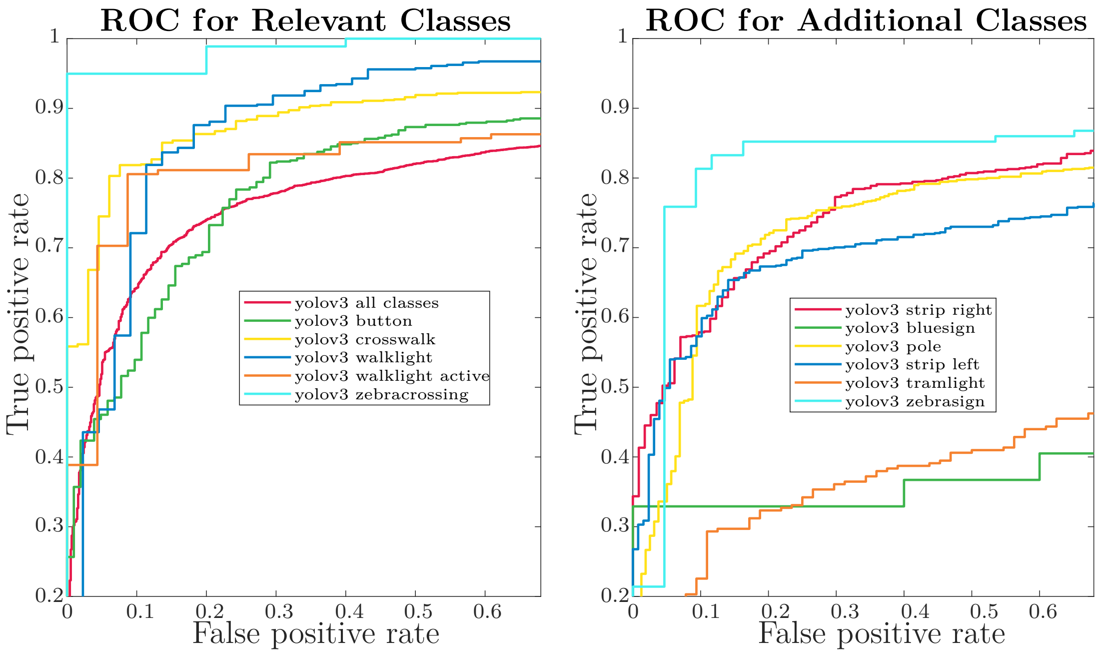


(a) (b)

Figure 4.12: Examples from our own dataset for pedestrian crossing detection that were labeled using our lab's labeling tool "sloth."

4.3.3 Evaluation

We evaluate our approach on a test split of our own dataset, containing separate videos that were labeled as well. After testing many different hyperparameters, *e.g.*, the input image size, number of layers, grid size, or number of anchor boxes, we report results for our YOLOv3-based network, trained as mentioned in the previous section. Our final model size is ∼250 MB for the default 107 layers of YOLOv3, and while training on our own dataset takes a few days, the system is capable of running at 11 frames per second for an input image size of 416x416 pixels, using a grid size of 13x13 and 5 anchor boxes. For this evaluation, as well as for the user study prototype (*cf.* Section 5.2) we use the following YOLO parameters: Our "objectness" threshold is 0.4, *i.e.*, YOLO internally uses a measure to decide for each region proposal whether there actually is an object contained at all. Furthermore, we use a hierarchical threshold of 0.5, a value that is used by YOLO when traversing class hierarchies, in order to determine how deep these trees should be followed, *i.e.*, how hierarchically specific a detection should be. As we don't yet rely on such a hierarchy for our very few classes, this parameter only affects us on our single level and thus degrades to our detection threshold, *i.e.*, all object candidates with a class probability higher than 0.5 are considered as a proper object detection. Finally, we use an Intersection over Union (IOU) value of 0.4, *i.e.*, when YOLO detects overlapping bounding boxes, this value is used in a non-maximum suppression step to remove the least confident object detections that overlap each other by more than this third parameter.

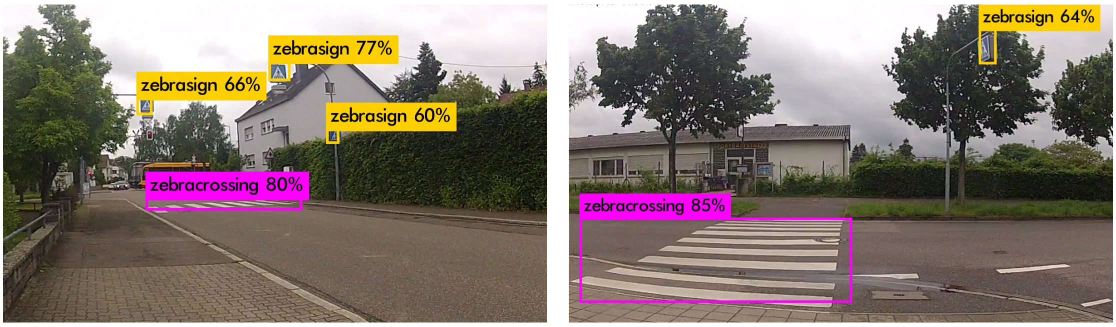


(a) (b)

Figure 4.13: ROC results for relevant pedestrian crossing classes (a) as well as additional ones (b). All relevant classes are detected with high accuracy, while decreased accuracy for other classes is mostly due to them being underrepresented in our training data.

Figure 4.13 shows our results for the most relevant pedestrian crossings classes, as well as some additional classes that, depending on a possible prototype implementation, could prove useful as well. For the main object classes, achieved accuracies are (mAP is the mean average precision): 98%mAP for zebra crossings, 83%mAP for traffic lights, 81%mAP for signal request pushbuttons, and 85%mAP overall — the overall score also includes the additional classes. We generally generate very accurate bounding boxes for almost all classes and detect relevant objects with a very high accuracy, especially zebra crossings, as these are sufficiently represented in our training data (∼1400 instances), given their rather straight forward appearance, a circumstance we also noticed in section 3.1.3 when detecting zebra crossings in aerial imagery.

Also very well detected, and classified, are the pedestrian traffic lights, slightly better for the don't walk phase light, *i.e.*, *red*, which is most likely also based on the simple fact that the data contains roughly twice as many of these, compared to the walking phase (∼3200 *vs.* ∼1600). A similar detection accuracy can be observed for the crosswalk section, *i.e.*, in our case this is a combination of the left and right



(a) (b)

Figure 4.14: Two example zebra crossing detections, one from further afar (a) and one close very up, short before crossing it (b). These are further discussed in the text.

pavement markings that indicate the crossing region for pedestrians. However, this accuracy is slightly skewed, as the crosswalk labels where generated by combining the left and right pavement markings, if either of both was available. Since a single marking, either left or right pavement, is already enough for the crosswalk section to be detected, its accuracy can potentially be much higher than those of the individual left and right pavement markings, which it actually is. The signal request buttons, blue zebra crossing signs (as found in Germany, these look completely different in other countries) and the traffic light poles are all detected adequately well. Only the tram line light, a red light that warns about an oncoming tram, and the street name signs, included only to improve the detection score of the zebra crossing signs, perform significantly worse, but this is most likely again caused by being insufficiently represented in our own dataset (*cf*. Table 4.3), and thus were not used for any further purposes or evaluations — although it very likely still helps discriminate other classes.

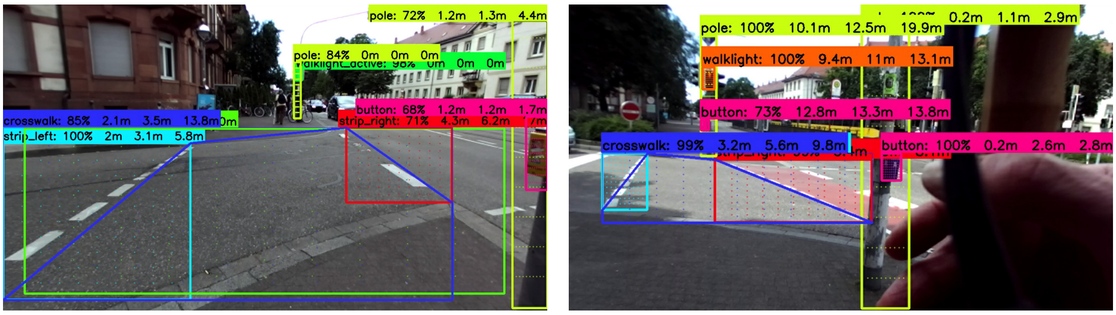
Qualitative Results

In addition to the presented numerical evaluation, we also want to discuss a few qualitative examples presented in figure 4.14 and figure 4.15 in a bit more detail.

Figure 4.14a shows that our approach, as suggested already by the numerical evaluation, is capable to detects zebra crossings even from afar. We also have included the German zebra crossing notification signs in our data, which are originally intended for drivers to slow down in case somebody wants to use the crossing as they have to as per German law, are nicely detected as well, albeit with a lower certainty than the zebra crossing itself. The main reason for also including this data is that very often these signs, although quite small, are long visible before the zebra crossing itself, often due to obstructions by other traffic participants. As we wanted to inform the user, and guide him towards, such zebra crossings as early as possible, we found these signs to be a reliable early indicator for an upcoming crossing, and can therefore notify a user about it at a very early stage.

Figure 4.14b presents a different zebra crossing, this time shortly before actually crossing it. While such a case is the one most relevant to people with visual impairments, as zebra crossings are approached this way, we found that when training our deep learning-based approach on other car-centric datasets, the detection score dropped significantly when this viewpoint was reached, up to a point where the zebra crossing was eventually not identified at all. This is clearly due to this particular perspective being highly underrepresented in those datasets. Interestingly, the zebra crossing indication sign is still detected, with almost the same score as before. We found out that this is only the case when training on the augmented data, where scaling along a single image axis could produce data instances similar to this one.

Figure 4.15a displays a pedestrian traffic light at a complicated throughabout, taken during our user study. This and the next image's locations where not part of our own training data, they had not been seen before by the network. The crosswalk region, a combination of left and right pavement markings, is almost correctly identified. Our approach is capable to detect most pedestrian traffic light related objects with high accuracy, while the provided distances represent: distance to closest point of object, median distance to bounding box, and maximum distance to object. We include these three separate distance measurements in our detections, because when we use these in a prototype, we often want to report different distances. For example, when approaching an object, such as a zebra crossing or pedestrian traffic light, it is important to know how far it is away. Once actually *on* the crossing itself, the minimal distance would always be the closest piece of ground perceived by the camera, while we are actually interested in the maximum distance, *i.e.*, the distance we are still required to travel until reaching the safe pedestrian walkway again. Please not that while this approach works reasonably well for ground markings, here it fails for the pedestrian traffic light and its pole, possibly due to a high noise level in the depth data, a case where we prefer to report the distance as zero, *i.e.*, the distance will be communicated to the user as "currently unknown."



(a) (b)

Figure 4.15: Two example pedestrian signal detections. These are further discussed in the text.

Figure 4.15b also shows a similar situation at the same throughabout, albeit on a pedestrian crossing prior to the first image. In this case, the distances for the opposing pedestrian are provided by the system, however for objects like traffic lights these three values should ideally be the same, if noise was not an issue. However, we found that even when parts of the background were included in the bounding box, causing its distance to be considered as well, relying on the median value provides sufficient accuracy for this and similar objects. Also visible in this specific image is, as mentioned elsewhere, that during the user study a user might partly cover the camera with its hand and the White Cane while waiting for a traffic light. As can be seen in this image, this does merely cause any issues. While the minimal distance for the signal request button is skewed from the hand, the median distance given is still accurate. The pedestrian traffic light, signal request pushbutton and both poles are detected with a score of 100%, the crosswalk almost as well, and even the small pushbutton on the other side of the crossing is already detected. In such a situation we can provide anti-veering guidance relying on many different objects, causing it to be highly robust during the entire crossing process.

Further qualitative evaluation of our proposed approach for pedestrian crossing assistance was conducted as part a larger user study (*cf*. Section 5.2), where the images in figure 4.15 were taken from as well, *i.e.*, they contain the actual detections by our system during the user study.

4.3.4 Conclusion

We have proposed a deep learning-based pedestrian crossing assistance system, specifically for pedestrian traffic lights and zebra crossings. As there is no comparable work available, also due to the unavailability of benchmark datasets, we can only present our own results. However, we already achieve a sufficient detection and classification accuracy, using an end-to-end trained network, that not only detects relevant objects for zebra crossings, but also for pedestrian traffic light crossings. We correctly detect and classify the most relevant objects, *i.e.*, signal request pushbuttons, pavement markings, and traffic lights. As it is a fully end-to-end trained system, it can also be used for other locations, where the appearance of zebra crossings (*cf*. Figure 2.2) or pedestrian traffic lights (*cf*. Figure 2.3) might differ significantly (*cf*. Section 2.3.3), of course only *iff* sufficient training data is available.

Our trained network is capable of running at 11 frames per second, but a further speedup could potentially save on precious battery time, as the GPU would be less busy and allow for longer testing times of our prototypes. While our network model size is currently only ∼250 MB, an even smaller network could significantly improve the usability of our proposed approach for mobile prototypes, as the latter allows us to use it on other integrated hardware platforms that might only contain a much smaller GPU memory, *i.e.*, smart phones or specialized mobile computing platforms. Moreover, detection of additional objects, *i.e.*, cars, pedestrians, cyclists or selected obstacle classes, might significantly increase the usefulness of such a system. Finally, class hierarchies should be considered, to improve the detection results for similar classes, *i.e.*, left and right pavement markings, or different types of traffic lights as well as their state.

In addition to our numerical evaluation, we also integrated this approach into a mobile prototype and further analyzed its usefulness as part of a user study with fifteen participants, presented in section 5.2.

Chapter 5

User Studies

> "Testing shows the presence, not the absence of bugs."
>
> – Edsger W. Dijkstra

One of this thesis' major motivations has always been to create assistive systems that provide a measurable benefit to people with visual impairments. Therefore, after individually evaluating the computer vision algorithms of our proposed systems using quantitative measures, we have also conducted a few qualitative user studies to analyze whether our proposed approaches provide real benefits and where they require further investigation and improvements, similar to the user studies analyzed in section 2.4, which we have drawn our key concepts from, provided in section 2.5.

Another important aspect to consider is that we never intended to command or instruct people when relying on our systems, but limit ourselves to inform and provide assistance where possible and required. People with visual impairments have to undergo extensive training with Orientation&Mobility specialists to acquire the necessary skills and the confidence for self-sufficient, truly independent mobility. We have to not only be careful not to contradict these acquired O&M techniques, but also to not limit their self-sufficiency by conveying that they must follow along a robot-like pre-planned path, unless, of course, they desire to do so, because the situation might require this. Therefore, a major implementation aspect in order to conduct user studies in the first place is the availability of a truly mobile prototype. It should always be as small and lightweight as possible, ideally hands-free, real time capable, have a long-lasting battery life, be unobtrusive to not raise suspicion by others, work reliably and not increase safety risks.

We have built a prototype to fulfill these constraints and discuss its hardware details in section 5.1. Finally, we present the conducted user study of our computer vision-based assistive systems in section 5.2 and finally discuss its results in section 5.2.3.

Our Contributions. We implement our proposed pedestrian crossing approach (Koester et al., 2019) presented in section 4.3 into a mobile prototype. Afterwards, we evaluate its assistive capabilities before and during the crossing period in a user study as part of the TERRAIN project in section 5.2.

Acknowledgements. This chapter contains joint work with Angela Constantinescu, who was essential in the organization and conduction of the TERRAIN project's user studies, in charge of the interface ideas and implementations, and thankfully always willing to discuss our crazy ideas for some of the proposed assistive systems.

5.1 Terrain Prototype

After trying out various hardware form factors, we settled on using an off-the-shelf laptop with an integrated GPU, capable to run deep learning methods. For this specific prototype, we used a "Gigabyte P35" laptop with a "NVIDIA GTX 1070" GPU built into it (*cf*. Figure 5.1). As laptops, and mobile units in general, only provide very limited processing power and battery time, especially when using the GPU, we limit the frame rate to five frames per second, adding a delay of roughly 200 milliseconds. To further reduce the battery consumption, the deep learning network's inference is only executed when requested by the user or the GNSS-based navigation, *e.g.*, when approaching a pedestrian crossing, causing the prototype to only idle around when not actively used. We use "Stereolabs[1] ZED" and "ZED Mini" cameras — both cameras provide a different depth range at a different size and weight factor.

We rely on the ZED's depth estimation to provide distance information required for recognized objects. This distance is especially important, as usually the most relevant object for immediate guidance, *i.e.*, considered pedestrian crossing aspect,

[1] https://www.stereolabs.com

Figure 5.1: The first TERRAIN prototype. It consists of the following components: (A) a modified backpack to allow for an improved airflow for the laptop inside, (B) an iPhone inside a sports armband for hands-free usage, (C) a ZED stereo camera mounted to the backpack's front straps, making it very easy to put on and take off, and (D) bone conduction headphones (Bluetooth). Furthermore, not labeled, but in the center of the image, an additional GPS receiver for logging purposes and a Bluetooth splitter to allow usage of both headphones at the same time (one for the participant and one for the supervisor).

is also located closest to the user. Therefore, in a post-processing step, we always consider only the nearest detected object of each class, *i.e.*, we discard other detections when there already exists a closer one. We must assume that this instance is much more relevant to the current situation faced by the user — it is only very rarely that this is not the case for our specific tested situations. The remaining object detections are sent to the main communication device of the system via Bluetooth Low Energy (BLE), *i.e.*, the iPhone, which then also relies on BLE to communicate with the user's preferred interface devices. We are currently investigating a second-generation prototype that uses much smaller processing components and hope to increase the battery lifetime significantly by further improving our algorithms, to create a less obtrusive system.

5.2 Terrain Project

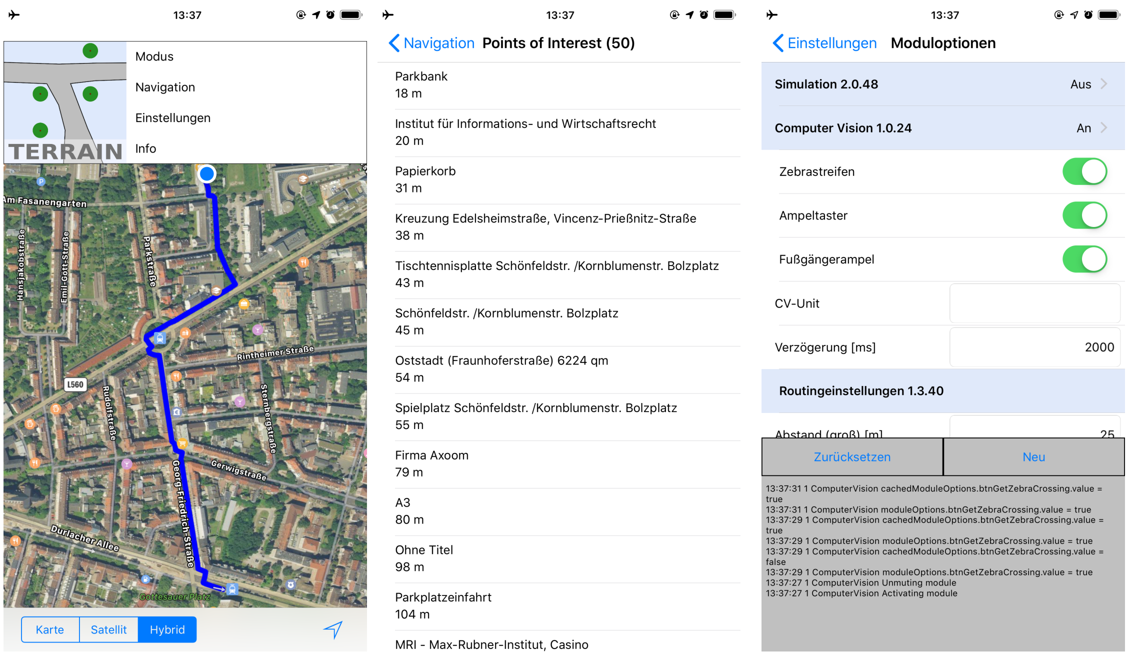
As part of the three-year project “Selbständige Mobilität blinder und sehbehinderter Menschen im urbanen Raum durch audio-taktile Navigation (TERRAIN),” funded by the “Bundesministerium für Bildung und Forschung”[2] (BMBF) between 2016 and 2019, we evaluate our computer vision-based assistance approach in combination with other O&M aspects. Originally, we intended to evaluate all of our approaches in a single user study, however, this plan had to be postponed, as it became clear in early experiments that a combination of all systems would overwhelm the participants and therefore only evaluate the pedestrian crossing system, while a combination of all modalities will be evaluated towards the very end of this project. The TERRAIN consortium consists of multiple partners, *i.e.*, the KIT's “Computer Vision for Human Computer Interaction Lab (CV:HCI)”[3], the KIT's “Study Center for the Visually Impaired (SZS)”[4] and the KIT's “Institute for Technology Assessment and Systems Analysis (ITAS)”[5], as well as the two involved companies iXpoint[6], and Papenmeier[7].

5.2.1 User Study

iXpoint developed the modular iOS-based smartphone application (*cf*. Figure 5.2), used as the navigation interface and central unit that connects all aspects, *e.g.*, Bluetooth Low Energy (BLE) bone conduction headphones, a mobile Braille unit, GNSS information and computer vision input. It provides encapsulated module environments, which are programmed in a domain specific language (Ritterbusch et al., 2018) and allows a concurrent development of all its aspects, *e.g.*, the routing, computer vision, and human-computer interaction model can all be improved separately and independently. The application furthermore allows a very detailed configuration of all connected devices and individual parts, where all aspects of the system can be individualized to personal requirements and preferences.

iXpoint also integrated a roadside aware routing approach based on OSM data (Ritterbusch and Kucharek, 2018) that focusses on roadside aware routes and pedestrian

[2] https://www.bmbf.de/
[3] https://cvhci.anthropomatik.kit.edu
[4] https://www.szs.kit.edu
[5] https://www.itas.kit.edu
[6] https://www.ixpoint.de
[7] https://www.papenmeier.de



(a) (b) (c)

Figure 5.2: iXpoint's iOS application, developed for TERRAIN as the main user interface: (a) the startup screen, a main menu and map area with an activated route, (b) a list of Points of Interest in the user's vicinity, (c) the settings dialog for the computer vision unit and other modules that were used during the user study.

sidewalk generation, but also searches for the safest possible path and used pedestrian crossings, similar to our own routing approach (*cf.* Section 3.2.1). We did not include our own routing approach, since iXpoint preferred the routing algorithm to rely on generated sidewalk information over shoreline-level routing, as such detailed data only has limited availability. However, for this user study, we relied on a fixed GNSS track that was generated from the algorithm, but making it fixed allows for less differences between participants and increases reproducibility, although such an approach does not allow for situation-based re-routing in case of errors.

The speech interface is similar to existing navigation systems, *i.e.*, short, precise and intuitive, as there was little training time per participant to get accustomed to these. Similar to the Wizard of Oz experiment (*cf.* Section 3.2.3), directions are conveyed using the clock-based system, *i.e.*, "eleven" refers to slightly left, "twelve" to straight ahead, "one" to slightly right, and so on, including "half" for directions in between (Sánchez and de la Torre, 2010). A possible sequence of speech instructions

[8] CC BY-NC-ND 2.0 https://creativecommons.org/licenses/by-nc-nd/2.0/

(a) (b) (c)

Figure 5.3: A participant during the TERRAIN user study: (a) on a pedestrian walkway, (b) in front of a zebra crossing, (c) crossing at a pedestrian traffic light. The backpack contains the prototype's processing unit, while the camera is mounted onto its front strap.

when approaching a zebra crossing could be: [*Nav*]: Follow sidewalk 42 meters, [*Nav*]: In 20 meters cross Kaiserstraße at zebra crossing, [CV]: Zebra at twelve in 7 meters, [CV]: Zebra at one in 4 meters, [CV]: Zebra at twelve in 1 meter, where *Nav* is the high-level routing information from the navigation system and CV low-level guidance-based on computer vision. Feedback was communicated using bone conduction headphones, although ongoing BLE issues forced us to also use a Bluetooth speaker.

Participants

All study participants exhibited severe visual impairments and were very interested in our technology. More precisely, the study consisted of fifteen participants with visual impairments, aged between 21 and 71 years: four blind from birth, two late blind, nine with severe visual impairments. Eight of them could neither perceive traffic lights nor zebra crossings, or just barely, while six were not able to distinguish traffic light phases, just zebra crossings very weakly when already very close. However, all participants where highly mobile and move around by walking, while eleven also rely on public transport. Only one participant always walks around alone, thirteen often, and one only rarely. Thirteen of the participants rely on additional navigation aids, some of them only in unknown areas and thirteen use the White Cane as well, while two did not require any aiding devices. The area of the user study was completely unknown to twelve of them, while three were aware of the neighborhood, but had

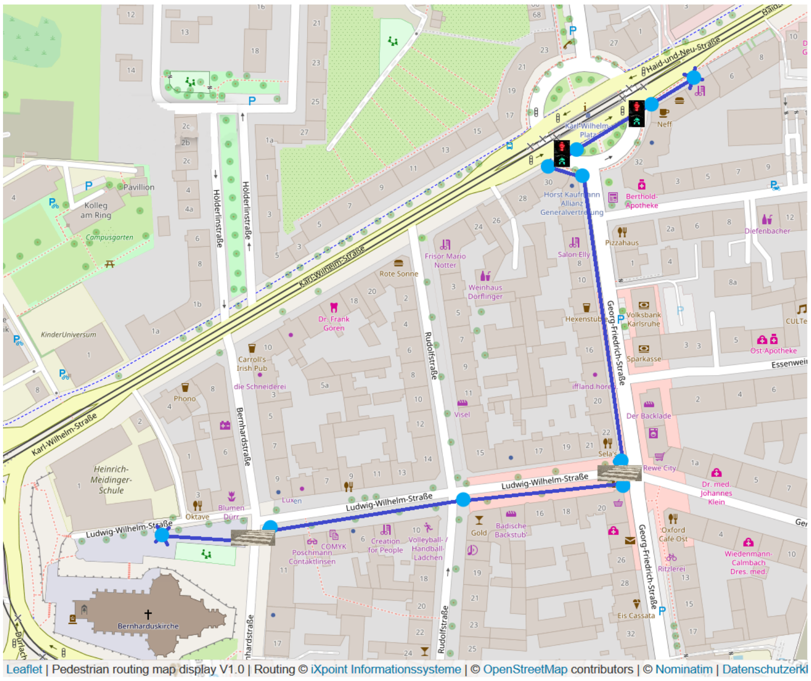


Figure 5.4: The test routes for the TERRAIN user study. Participants were asked to use our navigation and computer vision prototype to maneuver around in an unknown area. The selected routes contain multiple pedestrian crossings. © (iXpoint Informationssysteme)

not been in this specific location before. Figure 5.3 shows a participant wearing the prototype in three typical situations encountered during the user study.

Routes

We studied our system on two routes in a busy urban area, integrating traffic lights and zebra crossing (*cf*. Figure 5.4). Additionally, the path up to the first route was used for a short training session for the participants to get accustomed to the whole system and try the navigation as well as its computer vision modalities in a controlled setting. Overall, the participants evaluated the system on three different routes. The first route, $Route^T$, was the training route leading from our lab to the first start point and allowed participants to test the computer vision on a zebra crossing as well as a pedestrian traffic light. These could be crossed as often as desired. Part of the reason to rely on a fixed track was that it allowed to artificially increase the number of pedestrian crossings. Then on $Route^F$ the evaluation was started and on $Route^B$—same as $Route^F$ but backwards—the other modality evaluated on two zebra crossings as well as two pedestrian traffic lights, *i.e.*, conducted once with ($Route_{CV}$) and once without ($Route_{Nav}$) computer vision. Although $Route^F$ and $Route^B$ are practically equivalent, participants were not made aware of this fact beforehand. Eight of the participants walked first with the computer vision assistive system ($Route^F_{CV} \rightarrow Route^B_{Nav}$), the other seven used it only on the second route ($Route^F_{Nav} \rightarrow Route^B_{CV}$).

Participants were required to fill out questionnaires before, in-between, and after each route completion. Also, a sighted escort, *i.e.*, a mobility trainer or specially instructed volunteer, accompanied the participant for safety reasons, as well as the test instructor. A crossing procedure was implemented, for safety reason as well: participants would approach a pedestrian crossing as usual and position themselves for crossing, then notify the test instructor that they intended to do so, wait for the OK from the instructor — to assure there was no dangerous situation — and finally cross, closely followed by the escort and the instructor. Therefore, the participants still had to decide themselves where to walk, but should rely on the additional information provided by the computer vision prototype. On the $Route_{CV}$, the backpack including the prototype (*cf*. Figure 5.1) was carried by the user (*cf*. Figure 5.3), while on $Route_{Nav}$ it was usually carried by the test instructor to reduce physical exertion and allow subjects to also consider the added weight of the prototype in their evaluation. Furthermore, a poor GNSS signal often caused navigation instructions to be missed by the routing system, so the test instructor then manually read these out loud.

5.2.2 Evaluation

We gathered data from multiple sources, *i.e.*, questionnaires (participant backgrounds, evaluations, NASA-TLX, System Usability Scores and comments during the study), log files (by the iOS application, GPS tracks, items detected by computer vision, crashes, BLE issues), and videos — the prototype also recorded anonymized videos during its activation periods. We were especially interested, w.r.t. the computer vision assistive system, whether it provides a better orientation at crossings, increases perceived spatial awareness, provides sufficient accuracy and how much the perceived cognitive load had increased.

Computer Vision Prototype. The computer vision's deep learning methods had not used any training data from the routes taken during the user study, *i.e.*, it was all unseen data. Nonetheless, out of 65 encountered zebra crossings, it correctly

Table 5.1: The object detection performance of our computer vision prototype as observed in the video recordings during the user study. We report numbers on the total amount (#), true positives (TP), false negatives (FN), false positives (FP), recall and precision.

	#	TP	FN	FP	Recall	Precision
Zebra Crossings	65	62	3	6	95.4%	91.2%
Pedestrian Traffic Lights	138	127	11	8	92.0%	94.1%

identified 62 without interruptions, while three instances were not identified due to camera alignment issues — sometimes the camera would be aimed too far up. Also, only six false positive zebra crossings were detected during the entire study duration, but mostly for only a single frame and these could get filtered out easily. Furthermore, out of 138 pedestrian traffic light signals (including their state), 127 were correctly identified, while eight traffic lights for cars were wrongly detected as false positives, but for very brief moments only (*cf*. Table 5.1).

Orientation at Pedestrian Crossing. During the user study, we counted how well the participant found the crossing and awarded points, *i.e.*, 0 points when the crossing was not found or the participant did not properly align towards it, 0.5 points when the participant required a hint or a second attempt to properly align to it, and 1 point when the crossing was found and used successfully at the first attempt. The success rate in locating the crossing was not much improved on average, *i.e.*, 3.6 *vs.* 3.7 for the computer vision assistance, out of four crossings per participant, and therefore a maximum of four possible points. All participants were very mobile and already accustomed to crossing roads by themselves. In three cases participants even performed worse when using the computer vision prototype and misaligned themselves on the most difficult pedestrian crossing, due to finding the input from the system confusing in this specific instance.

Travel Time. Without the computer vision prototype, participants took 11 minutes on average, and with it 14.5 minutes. Our system made traveling slower, however we have to take frequent BLE issues into consideration, where the prototype had to be partly restarted to repair the connection issues (3.6 times per participant on average). The log files also suggest that the participants sometimes took additional time to parse the additional guidance input, which they did not require for their accustomed methods. We also noted a certain level of playfulness, where participants would await an extra phase at a pedestrian traffic light to experience the prototype communicating the current traffic light state in real time once more. Sadly, there was only a very short time to get accustomed to the system, while such effects likely wear off once participants have more experience with the prototype. However, as the most important aspect to be considered should always be safety, not just the travel time, we still consider this a success.

System Usability. The raw NASA-TLX ratings suggest only an insignificant increase in cognitive load, 23.6% and 24.3%. The System Usability Score was 69 without, and slightly lower at 63.5 when including the computer vision. Participants mostly complained about the rather heavy hardware (the full prototype weighs in at around 5 kilograms), and the repeated BLE connection issues. The order in which a participant evaluated both modalities had no visible effects on the individual scores, but we noticed that for participants where the system had less connection issues, the SUS scores were higher compared to those when not using the computer vision system.

Perceived Usefulness & Comments. Surprisingly, *all* fifteen participants preferred walking *with* the computer vision-based guidance system. Their overall prototype rating reflects this, *i.e.*, 1.7 for the computer vision assistance *vs.* 1.9 without it — on a one to five scale, with one being "very useful" and five "not at all useful." Ten of them were very intrigued by the prototype's ideas, especially the guidance potentially offered by computer vision-based approach. Furthermore, participants ($\mathcal{P}^{\#}$) commented positively on the following aspects: $\mathcal{P}^{9}$ said "Without the instructions I might not have found the zebra crossing." $\mathcal{P}^{7}$ "I loved the way it noticed traffic lights and zebras! Especially at zebras, one can orient oneself well." $\mathcal{P}^{13}$ "I felt more insecure without the [Computer Vision]." $\mathcal{P}^{15}$ "The location of zebras worked very well, I liked it very much." $\mathcal{P}^{2}$ "[I liked] that it announced traffic light state, especially where there's no push button." $\mathcal{P}^{8}$ "The traffic light state recognition." $\mathcal{P}^{14}$ "The traffic light state announcement." $\mathcal{P}^{12}$ "When it said: Zebra, push button, traffic light." At the same time, they also suggested some issues with our overall system, *i.e.*, they mostly disliked the GPS inaccuracies and the prototype's bulky and (for some participants) heavy hardware, the BLE issues and the time-taking necessary restarts this caused. The participants also suggested further objects to identify: $\mathcal{P}^{3}$ wished to be informed about obstacles in general, $\mathcal{P}^{6}$ about lampposts, $\mathcal{P}^{8,9}$ poles in general, $\mathcal{P}^{10}$ puddles, dog excrements and placards, and $\mathcal{P}^{13}$ about flat curbs.

5.2.3 Conclusion

While our proposed idea to rely on computer vision for additional guidance information was much appreciated and considered useful, our prototype needs further improvements in usability and reliability. First, it is possible to improve the GNSS accuracy, either by using additional systems, *i.e.*, Galileo, IMUs or even computer vision-based visual odometry. Second, the BLE connection issues need to be fixed, as they were a major drawback when relying on guidance information from the computer vision, which was not always immediately noticeable by the participants and the escorts. Third, the prototype was considered too heavy, especially by women, and the camera placement caused for some alignment issues with some participants — this should be made better adjustable in future systems.

While the computer vision-based object detection algorithm exhibited a very high accuracy, the distance estimation was often wrong, especially for one very wide pedestrian crossing that was used during the test route and that confused participants. Moreover, our prototypes battery only lasted for one hour while the deep learning-based model was running on its GPU, which was switched on and off manually by the test instructor when not in use. This should originally have happened GNSS-based, but the accuracy had always proven as way too unreliable.

While the time and accuracy measurements do not support the benefits of our approach, according to the participant's comments, the system did indeed help them to properly align before and during the crossing process. Even participants with sufficient vision to barely detect and locate the pedestrian crossings themselves, preferred to have the system for reassurance, especially for pedestrian traffic light phases or to locate signal request pushbuttons.

Interestingly, one participant suggested to integrate RTB's LOC.id[9], a BLE-based system that communicates traffic light state in real time and guides over the street. However, this is exactly the requirement we want to avoid, *i.e.*, to provide such assistance without requiring additional hardware to be installed at every single pedestrian crossing, but shows again just how much such information is valued.

Future work includes improvements of the computer vision's accuracy, to integrate everything on improved hardware with a better battery runtime, to reduce the weight

[9] https://www.rtb-bl.de/RTB/en/blind-aids/loc-id/technologyapplication/

and size of the entire prototype and to fix the BLE connection issues. Furthermore, we would like to include the other relevant suggested objects into our routing and guidance information as well.

Chapter 6

Conclusion

"The future has arrived — it's just not evenly distributed yet."

– William Gibson

In this thesis we have identified a spatial and systemic gap between low-level mobility techniques and high-level orientation instructions and addressed some of its far-ranging implications for people with visual impairments. We proposed computer vision-based techniques and approached this gap from both directions to better interconnect these two essentially different aspects — Orientation & Mobility (O&M). From a top-down perspective, we narrow this gap by improving high-level orientation through impairment aware navigation and a shoreline-level routing, as well as increase the availability of required data, *i.e.*, zebra crossing locations. Approached bottom-up, we further close this gap by improving low-level mobility through accessible section detection and inner shoreline guidance, as well as pedestrian crossing assistance. In accordance with our key concepts, we also suggested to integrate the intricate O&M techniques learned by people with visual impairment with generally available navigation systems, to prevent contradicting behavior that currently often occurs. We separated our contributions into the following major aspects:

- Key concepts used as guidelines for our own research efforts (Chapter 2);
- Impairment aware routing and geospatial data availability (Chapter 3);
- White Cane technique awareness and its integration into guidance (Chapter 4);
- A user study to evaluate actual benefits of one of our systems (Chapter 5).

6.1 Contributions

Aerial Imagery: detect zebra crossings for geospatial databases.
We presented a data driven approach to detect relevant objects for pedestrian routing visible in aerial imagery, in our case zebra crossings, in Koester et al. (2016). Combining OSM data and Google Maps aerial imagery, we automatically created a large, but imperfect, training corpus for a machine learning-based classifier. To reduce the amount of downloaded data and prevent spurious detections, we only considered actual road surface. Our algorithm is capable of detecting zebra crossings with a very high reliability, *i.e.*, with an error of less than one percent, and better generalize to new data. Furthermore, while other approaches relied on hand-engineered features, we can semi-automatically detect and correctly classify zebra crossings even across country borders, even using a classifier trained in a completely different region. Finally, we compared ourselves to the state of art and yield an improved precision at a similar recall, but require much less data compared to other works.

Impairment Aware Routing: shoreline-level navigation.
We suggested a novel impairment aware routing algorithm in Koester et al. (2017), based on a shoreline-level of detail, again modeled after an O&M technique. Using OSM data, we combined high-level navigation information, based on traditional routing approaches, with low-level guidance, based on shorelines. We therefore modeled the way that people with visual impairments maneuver around a place, *e.g.*, using the shorelining technique and avoiding to cross wide-open spaces if possible, and integrated these into our orientation information. Our routing adaptions furthermore allowed the creation of routes based on very specific, individually configurable, preferences w.r.t. to pedestrian crossings. The generated routes satisfied all of our pre-defined criteria, *e.g.*, they do not significantly increase travel distance, but increase the number of accessible pedestrian crossings. We tested our approach in a first Wizard of Oz experiment, due to insufficient GNSS accuracy, and received very encouraging comments from the test participant, especially w.r.t. to an increased spatial awareness and the usefulness of knowing about the next segment in advance.

Accessible Section Detection: detect free space and obstacles.
We investigated different methods to retrieve the accessible section using computer vision techniques. Our approach relies on true depth data, *i.e.*, point clouds, and uses an approach we named Slice&Dice, *i.e.*, it separates the point cloud into individual

slices, clusters the remaining points individually and then recombines relevant sections found into the overall accessible section. This approach transforms the problem of detecting the ground plane from three dimensions into a two-dimensional approach, using a few simple assumptions. The technique yields very precise results and allows for a real time capable accessible section detection. Furthermore, the dual problem of detecting obstacles is also considered — as a general lack of free space.

Shorelines: detect and precisely localize relevant shorelines.
We proposed a novel computer vision-based shoreline detection in Koester et al. (2018), based on an actual Orientation & Mobility technique when using the White Cane, *i.e.*, shorelining. Our algorithm slices depth data similar to the accessible section detection, acquired by a body worn camera, into individual layers and analyzes them separately, significantly increasing its robustness to errors and allowing for increased parallelism. It exhibits a very high precision, with errors of less than two degrees and a distance of only a few centimeters, for its detected shorelines and their individual segments. This very high precision allows a reliable forecast when a natural gap in a shoreline occurs, *i.e.*, a driveway in an urban area. Our approach is then able to precisely inform the user about the relative location and distance of the next shoreline segment. Having reliable information about upcoming segments increases spatial awareness and reduces stress for people with visual impairments, especially in unknown areas.

Pedestrian Crossing Assistance: increase accessibility and safety.
We created a computer vision-based pedestrian crossing assistance system in Koester et al. (2019) that detects and locates relevant objects. It is capable to detect signal request pushbuttons, pedestrian traffic lights and their state, crossing markers and zebra crossings, all in real time. We used a deep learning approach for the detection system and trained a network on our own data to locate the aforementioned objects. Our evaluation shows that the network exhibits a high accuracy for almost all of the relevant classes, where worse performing classes were simply underrepresented in our training datasets. While the existing approach already shows promising results, inclusion of further classes needs to be investigated and the overall accuracy increased.

User Study: evaluate proposed assistive system in real world situations.
We conducted a user study to evaluate the combination of roadside aware routing and pedestrian crossing assistance in Koester et al. (2019). Fifteen participants were requested to maneuver around in an unknown neighborhood, once without our

computer vision-based assistive system, as well as using it when crossing a street. While our chosen statistical indicators did not suggest any improvements when relying on additional crossing information provided by the computer vision prototype, the participant's comments suggested otherwise. All fifteen participants would prefer to own such a system, even though they all should be considered highly mobile, and perceived the provided information as very useful, once they got accustomed to it. Finally, we received many encouraging remarks to further investigate the development of this and similar systems that seamlessly combine high-level navigation and low-level guidance information.

6.2 Future Work & Open Research Questions

Multiple obvious issues and possible improvements have already been identified in the individual discussion sections: Truly mobile prototypes, to be used while traveling in unknown locations for a prolonged period of time, require much better integration, longer lasting batteries and less weight to be actually carried around in the future. It would also be very beneficial to integrate all systems into an everyday smartphone, only requiring the phone's internal processing memory, connectivity, and even already existing monocular camera with a sufficient battery time.

At the same time, the computer vision algorithms must be improved in all aspects, *i.e.*, more reliable detections, more object types, improved accessible section and obstacle detections, to detect other shoreline types as well, improve pedestrian crossing assistance, and everything in real time on the least powerful processing hardware possible — that is also affordable. However, while some of these seem possible within the next few years, safety considerations should always be of utmost importance, therefore a monocular camera might, at least for a considerable amount of time, not be able to provide the same guarantees as a true stereo setup that results in not perfect, but very reliable, depth data. Truly intuitive interfaces, which are modeled after real world techniques, also have to be further investigated, such as the White Cane bumping into the virtual shoreline segment — using haptic feedback — instead of relying on a pilot tone for the same guidance information. Finally, in order to use the full potential of computer vision-based techniques, an improved understanding of the surroundings at a higher level is very important.

An assistive system must not only know the route its user wishes to take, but should also understand the user's physical and mental limitations and constantly adapt accordingly. These factors are not only very individual, but also change over time, and even within a single route, *i.e.*, spatial awareness constantly fluctuates, which might lead to an increase in anxiety, especially when errors are made that cause the user to deviate from a known or pre-planned path. As each individual copes differently with these situations, limited by personal ability and experience, a truly helpful and widely applicable assistive system also has to account for such individualities.

To provide better guidance, an improved model of the environment is required, *e.g.*, street intersection layouts, available traffic lights, relevant obstacles and a basic understanding of traffic patterns and human behavior, to name a few, otherwise safety critical mistakes will continue happening, *i.e.*, to mistakenly report the state of an irrelevant traffic light. While this is currently being intensively investigated for autonomous driving and driver assistance, so far it has not really been considered for the guidance of people with visual impairments.

Ideally, we aspire for our proposed assistive technologies, whether they are based on computer vision or not, to someday provide the accuracy, reliability and acceptance to be integrated into Orientation&Mobility training at every possible stage. This could possibly allow for easier and faster training, greater autonomy and mobility, and a generally increased safety of people with visual impairments when mastering the challenges of today's urban traffic.

Daniel Koester

Personal Information

Born September 13th, 1984, Stralsund, Germany

Research Experience

2016–2019 **BMBF Project**.
"TERRAIN – Selbstständige Mobilität sehgeschädigter Menschen im urbanen Raum durch audio-taktile Navigation"

2014–2016 **BMBF Project**.
"AVVIS – Artificial Vision for Assisting Visually Impaired in Social Interaction"

2013 **Google Faculty Research Award**.
"A Mobility and Navigational Aid for Visually Impaired Persons"

2010 **Student Research Project**, *Harman Becker International*, Ittersbach, Germany.
"Adaptive Real Time Segmentation of Road Surface by means of Texture and Saturation"

2008-2010 **Student Assistant**, *KIT*, Humanoids and Intelligence Systems Lab.
German Research Foundation (DFG) Project: "Augmented Reality in the Operating Room"

2007-2008 **Student Assistant**, *KIT*, Institute for High Frequency Technology and Electronics.
Numerical field calculations, radio wave transmission through electromagnetically penetrable objects

Teaching

2013–2018 **Organizer & Supervisor**, *KIT*, Practical Course, Computer Vision for Human-Computer Interaction (Best Practical Course Award in 2015).

2014–2017 **Supervisor**, *KIT*, Seminar, Assistive Technologies for Visually Impaired Persons.

2016–2017 **Supervisor**, *KIT*, Seminar, Computer Vision for Human-Computer Interaction.

2008–2009 **Teaching Assistant**, *KIT*, System Architecture Group Tutor.

Education

2013–2019 **Doctoral Student, Computer Science**, *Karlsruhe Institute of Technology (KIT), CV:HCI (Computer Vision for Human-Computer Interaction Lab), Karlsruhe, Germany.*
"An Orientation & Mobility Aid for People with Visual Impairments"

2005–2013 **Diploma, Computer Science**, *Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.*
"A Guidance and Obstacle Evasion Software Framework for Visually Impaired People"

1995–2004 **General Qualification for University Entrance ("Abitur")**, *Johann Wolfgang Goethe Gymnasium, Stralsund, Germany.*

Language Skills

German native

English advanced *10+ years in high school, 1 year student exchange in Sacramento, California, USA*

French basic *5 years in high school*

Interests

Scientific Computer Vision, Human Computer Interaction, Image Processing, Machine Learning, Augmented Reality, Computer Graphics, Operating Systems and Security

Personal Sailing, Yacht Repair & Maintenance, Woodworking, Science Fiction

Löbertstraße 7 – 71665 Vaihingen an der Enz – Germany
+49 (721) 608 46266 • *daniel.koester@kit.edu*
https://cvhci.anthropomatik.kit.edu/~dkoester • *nilsonholger*

Publications

Daniel Koester, Boris Schauerte, and Rainer Stiefelhagen. Accessible Section Detection for Visual Guidance. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–6, July 2013. 78, 79, 83, 84, 85, 86

Manuel Martinez, Angela Constantinescu, Boris Schauerte, Daniel Koester, and Rainer Stiefelhagen. Cognitive Evaluation of Haptic and Audio Feedback in Short Range Navigation Tasks. In Klaus Miesenberger, Deborah Fels, Dominique Archambault, Petr Peňáz, and Wolgang Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 8548, pp. 128–135, July 2014.

Boris Schauerte, Daniel Koester, Manuel Martinez, and Rainer Stiefelhagen. Way to Go! Detecting Open Areas Ahead of a Walking Person. In Lourdes Agapito, Michael Bronstein, and Carsten Rother (eds), *Computer Vision - ECCV 2014 Workshops (ECCVW)*, *Lecture Notes in Computer Science*, vol. 8927, pp. 349–360, September 2014.

Daniel Koester, Björn Lunt, and Rainer Stiefelhagen. Zebra Crossing Detection from Aerial Imagery Across Countries. In Klaus Miesenberger, Christian Bühler, and Petr Peňáz (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 9759, pp. 27–34, July 2016. 9, 40, 43, 56, 120

Daniel Koester, Maximilian Awiszus, and Rainer Stiefelhagen. Mind the Gap: Virtual Shorelines for Blind and Partially Sighted People. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1443–1451, October 2017. 9, 27, 40, 56, 120

Manuel Martinez, Alina Roitberg, Daniel Koester, Boris Schauerte, and Rainer Stiefelhagen. Using Technology Developed for Autonomous Cars to Help Navigate Blind People. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 1424–1432, October 2017.

Daniel Koester, Tobias Allgeyer, and Rainer Stiefelhagen. Visual Shoreline Detection for Blind and Partially Sighted People. In Klaus Miesenberger and Georgios Kouroupetroglou (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 10897, pp. 123–131, July 2018. 10, 77, 121

Daniel Koester, Angela Constantinescu, Sebastian Ritterbusch, Daniel Höb, and Rainer Stiefelhagen. Where and How to Cross: An Assistive System to Improve Orientation & Mobility of People with Visual Impairments. ***Submitted to*** *ACM Transactions on Accessible Computing (TACCESS)*, 12(2), 2019. 10, 77, 108, 121

Bibliography

D. Ahmetovic, C. Bernareggi, and S. Mascetti. Zebralocalizer: Identification and Localization of Pedestrian Crossings. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pp. 275–284, 2011. 30, 50

D. Ahmetovic, C. Bernareggi, A. Gerino, and S. Mascetti. Zebrarecognizer: Efficient and Precise Localization of Pedestrian Crossings. In *22nd International Conference on Pattern Recognition (ICPR)*, pp. 2566–2571, 2014. 30

D. Ahmetovic, R. Manduchi, J. M. Coughlan, and S. Mascetti. Zebra Crossing Spotter: Automatic Population of Spatial Databases for Increased Safety of Blind Travelers. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '15, pp. 251–258, 2015. 18, 43, 49, 50, 51, 52, 53

C. Akinlar and C. Topal. EDLines: Real-Time Line Segment Detection by Edge Drawing (ED). In *18th IEEE International Conference on Image Processing (ICIP)*, pp. 2837–2840, 2011. 49, 90

H. Altwaijry, M. Moghimi, and S. Belongie. Recognizing Locations with Google Glass: A Case Study. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 167–174, 2014. 17

S. Andreev, J. Dibbelt, M. Nöllenburg, T. Pajor, and D. Wagner. Towards Realistic Pedestrian Route Planning. In G. F. Italiano and M. Schmidt (eds), *15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, *OpenAccess Series in Informatics (OASIcs)*, vol. 48, pp. 1–15, 2015. 20

J. Aranda and P. Mares. Visual System to Help Blind People to Cross the Street. In K. Miesenberger, J. Klaus, W. L. Zagler, and D. Burger (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 3118, pp. 454–461, 2004. 31

M. Avila Soto and M. Funk. Look, a Guidance Drone! Assessing the Social Acceptability of Companion Drones for Blind Travelers in Public Spaces. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, pp. 417–419, 2018. 23, 36

M. Avila Soto, M. Funk, M. Hoppe, R. Boldt, K. Wolf, and N. Henze. DroneNavigator: Using Leashed and Free-Floating Quadcopters to Navigate Visually Impaired Travelers. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, pp. 300–304, 2017. 23

Y. Bellik and R. Farcy. Comparison of Various Interface Modalities for a Locomotion Assistance Device. In K. Miesenberger, J. Klaus, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 2398, pp. 421–428, 2002. 24

A. Bhowmick, S. Prakash, R. Bhagat, V. Prasad, and S. M. Hazarika. IntelliNavi: Navigation for Blind Based on Kinect and Machine Learning. In M. N. Murty, X. He, R. R. Chillarige, and P. Weng (eds), *Multi-disciplinary Trends in Artificial Intelligence*, *Lecture Notes in Computer Science*, vol. 8875, pp. 172–183, 2014. 24, 35

J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. VizWiz: Nearly Real-time Answers to Visual Questions. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pp. 333–342, 2010. 3, 12

J. R. Blum, M. Bouchard, and J. R. Cooperstock. Spatialized Audio Environmental Awareness for Blind Users with a Smartphone. *Mobile Networks and Applications*, 18(3):295–309, 2013. 21

S. Bohonos, A. Lee, A. Malik, C. Thai, and R. Manduchi. Cellphone Accessible Information Via Bluetooth Beaconing for the Visually Impaired. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 5105, pp. 1117–1121, 2008. 16

R. R. A. Bourne, S. R. Flaxman, T. Braithwaite, M. V. Cicinelli, A. Das, et al. Magnitude, Temporal Trends, and Projections of the Global Prevalence of Blindness and Distance and Near Vision Impairment: a Systematic Review and Meta-Analysis. *The Lancet Global Health*, 5(9):e888–e897, 2017. 2

E. Brady, M. R. Morris, Y. Zhong, S. White, and J. P. Bigham. Visual Challenges in the Everyday Lives of Blind People. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 2117–2126, 2013. 12

T. Caldwell. On Finding Minimum Routes in a Network with Turn Penalties. *Commun. ACM*, 4(2):107–108, 1961. 58

J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 8(6):679–698, 1986. 47

E. Carrasco, E. Loyo, O. Otaegui, C. Fösleitner, M. Dubielzig, R. Olmedo, W. Wasserburger, and J. Spiller. ARGUS Autonomous Navigation System for People with Visual Impairments. In K. Miesenberger, D. Fels, D. Archambault, P. Peňáz, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 8548, pp. 100–107, 2014. 20

K.-Y. Chan, R. Manduchi, and J. M. Coughlan. Accessible Spaces: Navigating Through a Marked Environment with a Camera Phone. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '07, pp. 229–230, 2007. 22

R. Cheng, K. Wang, and S. Lin. Intersection Navigation for People with Visual Impairment. In K. Miesenberger and G. Kouroupetroglou (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 10897, pp. 78–85, 2018. 33

O. Chum and J. Matas. Optimal Randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(8):1472–1482, 2008. 92

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016. 99

C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3): 273–297, 1995. 6, 46

J. M. Coughlan and H. Shen. A Fast Algorithm for Finding Crosswalks Using Figure-Ground Segmentation. *2nd Workshop on Applications of Computer Vision, in conjunction with ECCV*, 5, 2006. 30

J. M. Coughlan and H. Shen. Terrain Analysis for Blind Wheelchair Users: Computer Vision Algorithms for Finding Curbs and Other Negative Obstacles. In M. Hersh (ed), *Proceedings of the Conference and Workshop on Assistive Technologies for People with Vision and Hearing Impairments: Assistive Technology for All Ages*, *CEUR-WS*, vol. 415, 2007. 27, 89

J. M. Coughlan and H. Shen. The Crosswatch Traffic Intersection Analyzer: A Roadmap for the Future. In K. Miesenberger, A. I. Karshmer, P. Peňáz, and W. L. Zagler (eds), *13th International Conference on Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 7383, pp. 25–28, 2012. 32

J. M. Coughlan and A. L. Yuille. The Manhattan World Assumption: Regularities in Scene Statistics Which Enable Bayesian Inference. In *Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS)*, NIPS '00, pp. 809–815, 2000. 83, 89

J. M. Coughlan, R. Manduchi, and H. Shen. Computer Vision-Based Terrain Sensors for Blind Wheelchair Users. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 4061, pp. 1294–1297, 2006. 27, 89

Á. Csapó, G. Wersényi, H. Nagy, and T. Stockman. A Survey of Assistive Technologies and Applications for Blind Users on Mobile Platforms: A Review and Foundation for Research. *Journal on Multimodal User Interfaces*, 9(4):275–286, 2015. 7, 14

N. Dahlbäck, A. Jönsson, and L. Ahrenberg. Wizard of Oz Studies: Why and How. In W. D. Gray, W. E. Hefley, and D. Murray (eds), *Proceedings of the 1st International Conference on Intelligent User Interfaces*, IUI, pp. 193–200, 1993. 56, 70

D. Dakopoulos and N. Bourbakis. Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40(1):25–35, 2010. 23

N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conferene on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005. 6, 45

E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, 1959. 54

W. Elmannai and K. Elleithy. Sensor-Based Assistive Devices for Visually-Impaired People: Current Status, Challenges, and Future Directions. *Sensors*, 17(3):565, 2017. 13

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 99

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 45

H. Fernandes, J. Faria, H. Paredes, and J. a. Barroso. An Integrated System for Blind Day-to-day Life Autonomy. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, pp. 225–226, 2011. 16

A. Fiannaca, I. Apostolopoulous, and E. Folmer. Headlock: A Wearable Navigation Aid That Helps Blind Cane Users Traverse Large Open Spaces. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '14, pp. 19–26, 2014. 17, 35

M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 91

G. Fusco, H. Shen, and J. M. Coughlan. Self-Localization at Street Intersections. In *Canadian Conference on Computer and Robot Vision (CRV)*, pp. 40–47, 2014a. 32

G. Fusco, H. Shen, V. Murali, and J. M. Coughlan. Determining a Blind Pedestrian's Location and Orientation at Traffic Intersections. In K. Miesenberger, D. Fels, D. Archambault, P. Peňáz, and W. L. Zagler (eds), *14th International Conference on Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 8548, pp. 427–432, 2014b. 32

S. Gallo, D. Chapuis, L. Santos-Carreras, Y. Kim, P. Retornaz, H. Bleuler, and R. Gassert. Augmented White Cane with Multimodal Haptic Feedback. In *2010 3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 149–155, 2010. 24

A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3D Reconstruction in Real-Time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968, 2011. 90

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 99

A. Geiger, M. Roser, and R. Urtasun. Efficient Large-Scale Stereo Matching. In *10th Asian Conference on Computer Vision (ACCV)*, *Lecture Notes in Computer Science*, vol. 6492, pp. 25–38, 2010. 78

M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007. 19

I. Gottlob, S. S. Wizov, and R. D. Reinecke. Head and Eye Movements in Children with Low Vision. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 234(6):369–377, 1996. 26

J. a. Guerreiro, D. Ahmetovic, K. M. Kitani, and C. Asakawa. Virtual Navigation for Blind People: Building Sequential Representations of the Real-World. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, pp. 280–289, 2017. 21, 35

V. Guillet, B. Rumpler, and J.-M. Pinon. Providing Help for Visually Impaired People's Navigation in an Urban Environment Using GPS. In K. Miesenberger, J. Klaus, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 2398, pp. 429–436, 2002. 19

R. Guy and K. Truong. CrossingGuard: Exploring Information Content in Navigation Aids for Visually Impaired Pedestrians. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 405–414, 2012. 20, 34

M. Haklay and P. Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. 19, 41, 57

K. Hara, S. Azenkot, M. Campbell, C. L. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R. H. Ng, and J. E. Froehlich. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, pp. 16:1–16:8, 2013a. 19

K. Hara, V. Le, and J. Froehlich. Combining Crowdsourcing and Google Street View to Identify Street-level Accessibility Problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 631–640, 2013b. 19

S. Hashino and S. Yamada. An Ultrasonic Blind Guidance System for Street Crossings. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 6180, pp. 235–238, 2010. 24

A. Helal, S. E. Moore, and B. Ramachandran. Drishti: An Integrated Navigation System for Visually Impaired and Disabled. In *Proceedings Fifth International Symposium on Wearable Computers*, pp. 149–156, 2001. 14

S. Hinz and A. Baumgartner. Automatic Extraction of Urban Road Networks from Multi-View Aerial Imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(1–2):83–98, 2003. 18

A. Hub, J. Diepstraten, and T. Ertl. Design and Development of an Indoor Navigation and Object Identification System for the Blind. *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*, (77-78):147–152, 2003. 25

V. Ivanchenko, J. M. Coughlan, W. Gerrey, and H. Shen. Computer Vision-based Clear Path Guidance for Blind Wheelchair Users. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '08, pp. 291–292, 2008a. 27, 89

V. Ivanchenko, J. M. Coughlan, and H. Shen. Detecting and Locating Crosswalks Using a Camera Phone. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1–8, 2008b. 30

V. Ivanchenko, J. M. Coughlan, and H. Shen. Crosswatch: A Camera Phone System for Orienting Visually Impaired Pedestrians at Traffic Intersections. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 5105, pp. 1122–1128, 2008c. 30

V. Ivanchenko, J. M. Coughlan, and H. Shen. Real-Time Walk Light Detection with a Mobile Phone. In K. Miesenberger, A. I. Karshmer, P. Peňáz, and W. L. Zagler (eds), *13th International Conference on Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 7383, pp. 229–234, 2010. 33

C. Jacquet, Y. Bellik, and Y. Bourda. A Context-Aware Locomotion Assistance Device for the Blind. In K. Miesenberger, J. Klaus, W. L. Zagler, and D. Burger (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 3118, pp. 438–445, 2004. 24

R. Jafri and M. M. Khan. Obstacle Detection and Avoidance for the Visually Impaired in Indoors Environments Using Google's Project Tango Device. In K. Miesenberger, C. Bühler, and P. Peňáz (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 9759, pp. 179–185, 2016. 24

J. José, M. Farrajota, J. Rodrigues, and J. du Buf. The Smart Vision Local Navigation Aid for Blind and Visually Impaired Persons. *International Journal of Digital Content Technology and its Applications*, 5:362–375, 2011. 25

J. S. Ju, E. Ko, and E. Y. Kim. EYECane: Navigating with Camera Embedded White Cane for Visually Impaired Person. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '09, pp. 237–238, 2009. 25

S. Kammoun, F. Dramas, B. Oriolaand, and C. Jouffrais. Route Selection Algorithm for Blind Pedestrian. In *International Conference on Control, Automation and Systems*, pp. 2223–2228, 2010. 15, 20

S. K. Kane, C. Jayant, J. O. Wobbrock, and R. E. Ladner. Freedom to Roam: A Study of Mobile Device Adoption and Accessibility for People with Visual and Motor Disabilities. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '09, pp. 115–122, 2009. 7

B. F. Katz, S. Kammount, G. Parseihian, O. Gutierres, A. Brilhault, M. Auvray, P. Truillet, M. Denis, S. Thorpe, and C. Jouffrais. NAVIG: Augmented Reality Guidance System for the Visually Impaired. *Virtual Reality*, 16(4):253–269, 2012. 15

J. F. Kelley. An Empirical Methodology for Writing User-friendly Natural Language Computer Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '83, pp. 193–196, 1983. 56, 70

J. F. Kelley. An Iterative Design Methodology for User-friendly Natural Language Office Information Applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984. 56, 70

A. Krizhevsky, S. Ilya, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger (eds), *Advances in Neural Information Processing Systems 25 (NIPS)*, NIPS' 12, pp. 1097–1105. Curran Associates Inc., 2012. 6, 98

M. C. Le, S. L. Phung, and A. Bouzerdoum. Pedestrian Lane Detection for the Visually Impaired. In *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pp. 1–6, 2012. 31

M. C. Le, S. L. Phung, and A. Bouzerdoum. Pedestrian Lane Detection in Unstructured Environments for Assistive Navigation. In *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2014. 31

T. Leung and G. Medioni. Visual Navigation Aid for the Blind in Dynamic Environments. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 579–586, 2014. 17

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (eds), *Computer Vision – ECCV 2014, Lecture Notes in Computer Science*, vol. 8693, pp. 740–755, 2014. 99

A. L. Liu, H. Hile, G. Borriello, P. A. Brown, M. Harniss, H. Kautz, and K. Johnson. Customizing Directions in an Automated Wayfinding System for Individuals with Cognitive Impairment. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '09, pp. 27–34, 2009. 34

J. M. Loomis, R. G. Golledge, R. L. Klatzky, J. M. Speigle, and J. Tietz. Personal Guidance System for the Visually Impaired. In *Proceedings of the First Annual ACM Conference on Assistive Technologies*, ASSETS '94, pp. 85–91, 1994. 14

D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pp. 1150–1157, 1999. 6

D. Luxen and C. Vetter. Real-time Routing with OpenStreetMap Data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pp. 513–516, 2011. 57

R. Manduchi, J. M. Coughlan, and V. Ivanchenko. Search Strategies of Visually Impaired Persons Using a Camera Phone Wayfinding System. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 5105, pp. 1135–1140, 2008. 22

R. Manduchi. Mobile Vision as Assistive Technology for the Blind: An Experimental Study. In K. Miesenberger, A. I. Karshmer, P. Peňáz, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 7383, pp. 9–16, 2012. 22, 34

R. Manduchi and J. M. Coughlan. The Last Meter: Blind Visual Guidance to a Target. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 3113–3122, 2014. 22

R. Manduchi, S. Kurniawan, and H. Bagherinia. Blind Guidance Using Mobile Computer Vision: A Usability Study. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pp. 241–242, 2010. 22, 34

J. M. S. Martínez and F. E. Ruiz. Stereo-based Aerial Obstacle Detection for the Visually Impaired. In *Workshop on Computer Vision Applications for the Visually Impaired*, 2008. 25

S. Mascetti, D. Ahmetovic, A. Gerino, C. Bernareggi, M. Busso, and A. Rizzi. Robust Traffic Lights Detection on Mobile Devices for Pedestrians with Visual Impairment. *Computer Vision and Image Understanding*, 148:123–135, 2016a. 32

S. Mascetti, D. Ahmetovic, A. Gerino, C. Bernareggi, M. Busso, and A. Rizzi. Supporting Pedestrians with Visual Impairment During Road Crossing: A Mobile Application for Traffic Lights Detection. In K. Miesenberger, C. Bühler, and P. Peňáz (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 9759, pp. 198–201, 2016b. 33

B. Matthews, D. Hibberd, and O. Carsten. Road and Street Crossings for Blind and Partially Sighted People: The Importance of Being Certain. Technical report, Institute for Transport Studies, University of Leeds, 2014. 5, 28, 39, 42, 55, 60, 96

S. Mattoccia and P. Macrı'. 3D Glasses as Mobility Aid for Visually Impaired People. In L. Agapito, M. M. Bronstein, and C. Rother (eds), *Computer Vision - ECCV 2014 Workshops*, *Lecture Notes in Computer Science*, vol. 8927, pp. 539–554, 2015. 25

B. Mayerhofer, B. Pressl, and M. Wieser. ODILIA - A Mobility Concept for the Visually Impaired. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 5105, pp. 1109–1116, 2008. 15

T. J. Mehigan and I. Pitt. Harnessing Wireless Technologies for Campus Navigation by Blind Students and Visitors. In K. Miesenberger, A. I. Karshmer, P. Peňáz, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 7383, pp. 67–74, 2012. 16

N. Molton, S. Se, J. Brady, D. Lee, and P. Probert. A Stereo Vision-Based Aid for the Visually Impaired. *Image and Vision Computing*, 16(4):251–263, 1998a. 29

N. Molton, S. Se, D. Lee, P. Probert, and M. Brady. Robotic Sensing for the Guidance of the Visually Impaired. In A. Zelinsky (ed), *Field and Service Robotics*, pp. 223–230, 1998b. 29

V. N. Murali and J. M. Coughlan. Smartphone-Based Crosswalk Detection and Localization for Visually Impaired Pedestrians. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–7, 2013. 32

G. Máttyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing Road Maps by Parsing Aerial Images Around the World. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1689–1697, 2015. 18

J. Na. The Blind Interactive Guide System Using RFID-Based Indoor Positioning System. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 4061, pp. 1298–1305, 2006. 16

M. Nishikiri, T. Sakai, H. Kudo, T. Matsumoto, Y. Takeuchi, and N. Ohnishi. System Supporting Independent Walking of the Visually Impaired. In K. Miesenberger, C. Bühler, and P. Peňáz (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 9759, pp. 186–189, 2016. 31

T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 6, 45

M. Okamoto, J. Akita, K. Ito, T. Ono, and T. Takagi. CyARM – Interactive Device for Environment Recognition Using a Non-visual Modality. In K. Miesenberger, J. Klaus, W. L. Zagler, and D. Burger (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 3118, pp. 462–467, 2004. 23

T. Ono, T. Komatsu, J.-i. Akita, K. Ito, and M. Okamoto. CyARM: Interactive Device for Environment Recognition and Joint Haptic Attention Using Non-visual Modality. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 4061, pp. 1251–1258, 2006. 24

M. Owczarek, P. Skulimowski, and P. Strumillo. Sound of Vision – 3D Scene Reconstruction from Stereo Vision in an Electronic Travel Aid for the Visually Impaired. In K. Miesenberger, C. Bühler, and P. Peňáz (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 9759, pp. 35–42, 2016. 26

V. Pradeep, G. Medioni, and J. Weiland. Robot Vision for the Visually Impaired. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 15–22, 2010. 25

B. Pressl and M. Wieser. A Computer-Based Navigation System Tailored to the Needs of Blind People. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 4061, pp. 1280–1286, 2006. 15

L. H. Quam. Road Tracking and Anomaly Detection in Aerial Imagery. Technical report, SRI International Menlo Park CA Artificial Intelligence Center, 1978. 18, 41

R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm. USAC: A Universal Framework for Random Sample Consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):2022–2038, 2013. 92

J. Redmon. Darknet: Open Source Neural Networks in C. `https://pjreddie.com/darknet/`, 2013–2016. 98

J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017. 98, 99

J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. arXiv 1804.02767, Cornell University, 2018. 98

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016. 98

S. Resnikoff, D. Pascolini, D. Etya'alé, I. Kocur, R. Pararajasegaram, G. P. Pokharel, and S. P. Mariotti. Global Data on Visual Impairment in the Year 2002. Bulletin of the World Health Organization 82:844–851, WHO, 2004. 2

M. T. Rice, A. O. Aburizaiza, R. D. Jacobson, B. M. Shore, and F. I. Paez. Supporting Accessibility for Blind and Vision-Impaired People With a Localized Gazetteer and Open Source Geotechnology. *Transactions in GIS*, 16(2):177–190, 2012. 20

S. Ritterbusch and H. Kucharek. Robust and Incremental Pedestrian Path Network Generation on OpenStreetMap for Safe Route Finding. In K. Miesenberger and G. Kouroupetroglou (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 10897, pp. 302–309, 2018. 110

S. Ritterbusch, P. Frank, and V. Petrausch. Modular Human-Computer Interaction Platform on Mobile Devices for Distributed Development. In K. Miesenberger and G. Kouroupetroglou (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 10897, pp. 75–78, 2018. 110

D. A. Ross and B. B. Blasch. Wearable Interfaces for Orientation and Wayfinding. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, ASSETS '00, pp. 193–200, 2000. 14

D. A. Ross and A. Lightman. Talking Braille: A Wireless Ubiquitous Computing Network for Orientation and Wayfinding. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '05, pp. 98–105, 2005. 15

J. Roters, X. Jiang, and K. Rothaus. Recognition of Traffic Lights in Live Video Streams on Mobile Devices. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(10):1497–1511, 2011. 33

J. Sánchez and N. de la Torre. Autonomous Navigation Through the City for the Blind. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pp. 195–202, 2010. 15, 72, 111

J. Sánchez, L. Guerrero, M. Sáenz, and H. Flores. A Model to Develop Videogames for Orientation and Mobility. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 6180, pp. 296–303, 2010. 22

D. Sato, U. Oh, K. Naito, H. Takagi, K. Kitani, and C. Asakawa. NavCog3: An Evaluation of a Smartphone-Based Blind Indoor Navigation Assistant with Semantic Features in a Large-Scale Environment. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, pp. 270–279, 2017. 16, 36

B. Schmitz. *Navigation Systems for Special User Groups*. PhD thesis, University of Stuttgart, Stuttgart, Germany, 2015. 21

B. Schmitz and T. Ertl. Individualized Route Planning and Guidance Based on Map Content Transformations. In K. Miesenberger, D. Fels, D. Archambault, P. Peňáz, and W. L. Zagler (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 8548, pp. 120–127, 2014. 20, 55

J. Schneider and T. Strothotte. Constructive Exploration of Spatial Information by Blind Users. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, ASSETS '00, pp. 188–192, 2000. 21

T. Schwarze, M. Lauer, M. Schwaab, M. Romanovas, S. Böhm, and T. Jürgensohn. An Intuitive Mobility Aid for Visually Impaired People Based on Stereo Vision. In *2015 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 409–417, 2015. 26

S. Se. Zebra-Crossing Detection for the Partially Sighted. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 211–217, 2000. 29

L. Shangguan, Z. Yang, Z. Zhou, X. Zheng, C. Wu, and Y. Liu. CrossNavi: Enabling Real-time Crossroad Navigation for the Blind with Commodity Phones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 787–798, 2014. 30

T. Shiose, K. Ito, and K. Mamada. The Development of Virtual 3D Acoustic Environment for Training 'Perception of Crossability'. In K. Miesenberger, J. Klaus, W. L. Zagler, and D. Burger (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 3118, pp. 476–483, 2004. 22

T. Shioyama, H. Wu, and N. Nakamura. Measurement of the Length of Pedestrian Crossings and Detection of Traffic Lights from Image Data. *Measurement Science and Technology*, 13(9):1450–1457, 2002. 29

J. L. Souman, I. Frissen, M. N. Sreenivasa, and M. O. Ernst. Walking Straight into Circles. *Current Biology*, 19(18):1538–1542, 2009. 12

B. Sucu and E. Folmer. The Blind Driver Challenge: Steering Using Haptic Cues. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '14, pp. 3–10, 2014. 22

J. M. Sáez, F. Escolano, and M. A. Lozano. Aerial Obstacle Detection With 3-D Mobile Devices. *IEEE Journal of Biomedical and Health Informatics*, 19(1):74–80, 2015. 26

H. Tatsumi, Y. Murai, N. Nagai, and M. Miyakawa. Marking in the Surroundings by Data-Carriers for the Visually Impaired. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 4061, pp. 1259–1264, 2006. 15

Y. Tian, X. Yang, and A. Arditi. Computer Vision-Based Door Detection for Accessibility of Unfamiliar Environments to Blind Persons. In K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer (eds), *Computers Helping People with Special Needs (ICCHP)*, *Lecture Notes in Computer Science*, vol. 6180, pp. 263–270, 2010. 17

M. S. Uddin and T. Shioyama. Measurement of the Length of Pedestrian Crossings - A Navigational Aid for Blind People. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 690–695, 2004. 30

M. S. Uddin and T. Shioyama. Bipolarity and Projective Invariant-Based Zebra-Crossing Detection for the Visually Impaired. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, p. 22, 2005. 30

S. Utcke. Grouping Based on Projective Geometry Constraints and Uncertainty. In *Sixth International Conference on Computer Vision*, pp. 739–746, 1998. 29

T. Völkel and G. Weber. RouteCheckr: Personalized Multicriteria Routing for Mobility Impaired Pedestrians. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '08, pp. 185–192, 2008. 19

M. Weber, P. Wolf, and J. M. Zöllner. DeepTLR: A Single Deep Convolutional Network for Detection and Classification of Traffic Lights. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 342–348, 2016. 31

W. R. Wiener, R. L. Welsh, and B. B. Blasch. *Foundations of Orientation and Mobility: History and Theory*, volume I. AFB Press, American Foundation for the Blind, third edition, 2010. 1, 26

M. A. Williams, A. Hurst, and S. K. Kane. Pray Before You Step out: Describing Personal and Situational Blind Navigation Behaviors. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, pp. 28:1–28:8, 2013. 13

M. A. Williams, C. Galbraith, S. K. Kane, and A. Hurst. "Just Let the Cane Hit It": How the Blind and Sighted See Navigation Differently. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 217–224, 2014. 13

S. Winter. Modeling Costs of Turns in Route Planning. *GeoInformatica*, 6(4): 345–361, 2002. 58

Z. Wu, W. Fu, R. Xue, and W. Wang. A Novel Line Space Voting Method for Vanishing-Point Detection of General Road Images. *Sensors*, 16(7), 2016. 90

R. Yang, S. Park, S. R. Mishra, Z. Hong, C. Newsom, H. Joo, E. Hofer, and M. W. Newman. Supporting Spatial Awareness and Independent Wayfinding for Pedestrians with Visual Impairments. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '11, pp. 27–34, 2011. 21

K. Yatani, N. Banovic, and K. Truong. SpaceSense: Representing Geographical Information to Visually Impaired People Using Spatial Tactile Feedback. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 415–424, 2012. 21

L. Zeng, D. Prescher, and G. Weber. Exploration and Avoidance of Surrounding Obstacles for the Visually Impaired. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '12, pp. 111–118, 2012. 24, 35

L. Zeng, M. Simros, and G. Weber. Camera-based Mobile Electronic Travel Aids Support for Cognitive Mapping of Unknown Spaces. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, pp. 8:1–8:10, 2017. 24

C. Zimring and J. Templer. Wayfinding and Orientation by the Visually Impaired. *Journal of Environmental Systems*, 13(4):333–352, 1983. 27