WASEDA UNIVERSITY

DOCTORAL THESIS

Map-less Navigation for Blind People

Masaki Kuribayashi

Graduate School of Advanced Science and Engineering
Department of Pure and Applied Physics
Research on Image Processing

March 15, 2026

Abstract

Map-less Navigation for Blind People

by Masaki Kuribayashi

This dissertation presents *map-less* navigation systems that assist blind people without relying on pre-prepared environmental information, to enable scalable deployment across diverse environments. While existing assistive navigation systems provide turn-by-turn guidance, their deployment remains limited because they require prebuilt maps and infrastructure, which incur substantial costs for creation and maintenance. Aiming for the future where blind users would be able to use assistive navigation systems in various places, we aim to overcome this scalability issue by removing the necessity of prebuilt maps and infrastructure.

To this end, we propose four systems: two smartphone systems, Corridor-Walker and Snap&Nav, and two robot systems, PathFinder and WanderGuide. However, removing maps introduces challenges, as the system does not know the environment in advance, where to guide the user, or what kind of information needs to be detected and conveyed. This contrasts with map-based systems, which can determine all of these aspects beforehand. To address this, we adopt collaborative interaction, where users actively engage in the system's sensing and decision-making processes to complement the capability limitations that arise from the map-less setting.

Two smartphone systems, Corridor-Walker and Snap&Nav, adopted active scanning by users so that the system can provide richer environmental information about intersections, enabling more accurate navigation. Snap&Nav extends collaboration beyond the system and user to include sighted assistants, who contribute by capturing a floor map to obtain route information. For the robot system PathFinder, we adopted an interaction model inspired by guide dogs, where the robot takes over mobility while the user participates in decision-making processes such as determining the direction to proceed. We enhanced this capability by incorporating functionalities such as sign recognition, which was designed through a participatory study. Furthermore, we extend the potential of a map-less system to exploration tasks by utilizing a multimodal large language model (MLLM) to describe the surrounding environment. Through participatory design, we developed functionalities to enhance exploration, such as a revisiting feature that allows users to return to previously visited locations. This enables users to identify places of interest based on MLLM-generated descriptions and visit them again. Finally, this dissertation developed the visual language navigation (VLN) benchmark Memory-Maze, which mimics scenarios where a robot navigates to the destination based on instructions provided by sighted passersby. Memory-Maze enables the robot to assist the user's decision-making process, which occurs within collaborative interaction.

Through user studies, we found that these map-less navigation systems increased confidence and reduced cognitive load compared to traditional aids, with participants accepting longer travel times in exchange for greater independence and broader applicability. Based on these results, situate our findings within the field of human-computer interaction, for example, the balance between a rich experience and task completion time. We also discuss future directions for our system, such as active participation in the control of the robot by utilizing VLN technology.

Acknowledgements

This thesis was supported by my advisors, coworkers, researchers, colleagues, and laboratory members. I extend my sincere thanks to all individuals who influenced my work.

I express my deepest gratitude to my advisor, Prof. Shigeo Morishima, who welcomed me as a student in his lab six years ago. Without Morishima-sensei's support, I could not have overcome the challenges along the journey toward obtaining my Ph.D. These six years at Morishima Lab. have been the most impactful and priceless period of my life, which will continue to shape my future. I also thank all the dissertation committee members, Prof. Hideyuki Sawada and Dr. Chieko Asakawa, for the thoughtful feedback on my work.

Secondly, I would like to express my sincere gratitude to Dr. Chieko Asakawa and Dr. Hironobu Takagi for their invaluable guidance in accessibility research. Asakawa-san has continually inspired me, and her lived perspective as a blind researcher has shaped the direction of my work. Asakawa-san provided me with numerous opportunities and pushed me to work on my best performance. Takagi-san has been both an outstanding manager and mentor, supporting my research through collaborations with my university, IBM Research Japan, and the Miraikan Accessibility Lab. My work would not have been possible without his advice and discussions.

I also extend my thanks to Dr. Seita Kayukawa, who initially guided me into the field of accessibility research. Beginning with the development of LineChaser alongside Kayukawa-san, I discovered a field that became my true passion, motivating me to pursue this path from undergraduate studies through to my doctoral degree. Because of you, I was able to maintain high standards throughout the journey.

I thank my co-authors and collaborators, who have influenced my Ph.D. work, including Jayakorn Vongkulbhisal, Tatsuya Ishihara, Daisuke Sato, Masayuki Murata, Kohei Uehara, Allan Wang, Renato Ribiero, Rayna Hata, Yaxin Hu, Yusuke Miura, Masaya Kubota, Yuka Kaniwa, Yuta Taguchi, and Karnik Ram. You all shaped my work.

I thank all Miraikan staff members, particularly the members from the Miraikan Accessibility lab., including Xiyue Wang, Ayaka Tsutsui, Kengo Tanaka, Kazuhiko Sugano, Hiromi Kurokawa, and Takashi Suzuki, for their support and discussion. Miraikan was the most enjoyable place to discuss ideas and work together.

I would also like to thank all members and friends in Morishima-lab for their respectful support. Thank you - Satoko Matsuda, Yoshihiro Fukuhara, Ayano Kaneda, Feng Qi, Hideki Tsunashima, Yoshiki Kubotani, Kazuhito Sato, Shinei Arakawa, Tomoya Yoshinaga, Ryosuke Oshima, Sara Kashiwagi, Kaoru Sasaki, Shunsuke Mitsumori, Hiroki Nishizawa, Shonoshin Sakamoto, and Arata Ito.

I want to express my sincere gratitude to my family members, Masahiro, Sachiko, and Yuuki, for their continuous support and encouragement. Finally, I extend my thanks to my wife, Momoko, for supporting my life, as well as my Ph.D. journey.

Works in this thesis were supported by the JST-Mirai Program (JPMJMI19B2) and JSPS KAKENHI (JP20J23018, JP23KJ2048, and 21H05054). I would also like to express my gratitude to the following organizations for their valuable support: Miraikan — The National Museum of Emerging Science and Innovation, the Consortium for Advanced Assistive Mobility Platform, and Mori Building Co., Ltd. Finally, I extend my sincere thanks to all participants in the user study.

Contents

Chapter 1

Introduction

1.1 Background

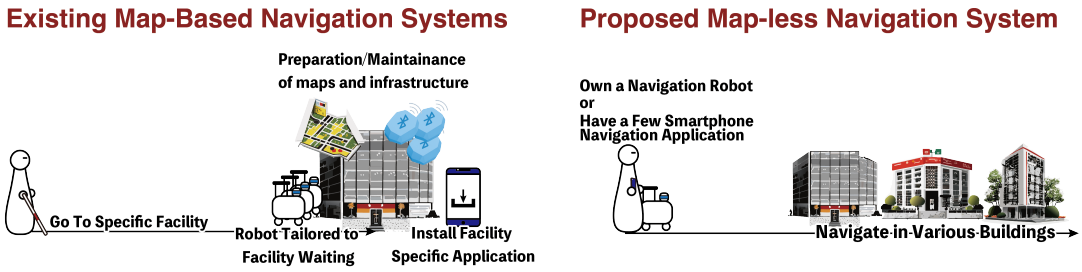


FIGURE 1.1: **Map-less Navigation System Vision.** Existing map-based systems are customized to specific facilities due to the need for infrastructure and pre-built maps. As a result, smartphone users must install different applications for different locations, and robots are typically deployed only within a single building. This robot deployment business model is referred to as the rental model [1]. Such solutions are therefore available only in limited facilities. In contrast, we envision map-less navigation systems that can be used across various buildings. For smartphones, users will have multiple applications, including well-established ones such as Google Maps, as well as a map-less navigation application that supports diverse indoor environments. For robots, we envision a future in which users own personal robots and navigate across different buildings.

Blind people face various challenges while navigating independently. The challenges during navigation can be grouped into three main areas: mobility, orientation, and spatial understanding [2]. Mobility is the ability to navigate environments safely and efficiently, such as avoiding surrounding obstacles. While canes are used to detect obstacles, blind people may still experience collisions, which can cause injury or damage to the environment (*e.g.*, knocking over a flower bin) and to others. Secondly, orientation is the ability to be able to locate current orientation and position in an environment so that they can determine the way to their destination. For blind people, orientation requires the use of auditory, olfactory, and tactile cues while referring to the mental map they have. This process poses a particularly high cognitive load. They could also miss a turn or fail to locate a landmark, which can cause misalignment between the map and their actual position, leaving them disoriented and lost. Third, the inability to visually perceive environments restricts spontaneous information obtaining [3]. Obtaining environmental information is essential not only for constructing a mental map, but also for discovering new information based on their interest. Knowing what is in the surroundings also relates to the orientation challenge, as orientation is also determined by establishing landmarks as reference points. Despite these challenges, many blind people express a strong desire to navigate independently [4].

Therefore, many academic and industrial efforts have developed navigation systems for blind people [5, 6, 7]. These systems typically use prebuilt maps and infrastructure such as Bluetooth Low Energy (BLE) beacons, enabling precise localization for turn-by-turn guidance to the predetermined destination [8, 9, 10, 11]. Those would include off-the-shelf smartphones (*e.g.*, BlindSquare [12], NavCog [9], and ChitChatGuide [13]) and customized wearable systems (*e.g.*, ISANA [10]) that assist orientation and spatial learning by providing turn-by-turn guidance while describing surrounding information. As with these systems, users still have to handle their own mobility, recent research trends have shown particular interest in robotics systems as they could autonomously navigate blind users [14, 15, 16, 11]. This allows blind users to focus more on orientation and perception of their surroundings [17], extending the scope of usage beyond navigation to tasks such as exploration [18].

However, most of these assistive smartphone, wearable, robotic, and *etc.* systems cannot be used in various places and face a scalability problem, as they typically rely on infrastructures and prebuilt maps. For example, these systems often require infrastructure such as BLE beacons for localization, along with static route maps that describe topological connections between points and sometimes environmental landmarks. Also, in the case of robotic systems, light detection and ranging (LiDAR) maps created through 3D scanning of the environment to represent static and dynamic obstacles are required. Although floor maps (*i.e.*, a map of a floor which is typically placed in the entrance of a building) or map information on the internet exist, they differ from the digital maps required by these assistive systems. Converting such information into the precise, system-compatible digital format typically requires specialized development with technical experts. Therefore, the substantial costs associated with creating, deploying, and maintaining maps and infrastructure often make building managers reluctant to invest, resulting in sparse deployment across public buildings. Consequently, existing systems remain unscalable, leaving blind people without reliable navigation assistance in most of the buildings they encounter.

In addition, the requirement of maps and infrastructure [17] poses a particular limitation for the real-world deployment of robotic systems. Because of these requirements, the current dominant business model for deploying such systems is the *rental model*, in which robots remain stationed at specific facilities to serve users on-site (*e.g.*, the Science Museum [19]) [1]. To realize a future in which users can navigate across various locations by potentially owning a robotic system, these systems must operate without reliance on infrastructure and maps.

Therefore, this dissertation presents a *map-less* navigation system that operates without any predetermined environmental information, such as prebuilt maps or external infrastructure (Figure 1.1). The purpose of map-less navigation systems is to enable blind users to travel independently, even in new or unfamiliar environments, whether for the system itself or for both the user and the system. The map-less system in this dissertation is specifically designed for indoor environments. For example, consider a scenario where a user wants to visit a building they have never entered before. The user can reach the building entrance using existing outdoor navigation systems such as Google Maps, in which typically has updated map information. Inside buildings, the system often lacks access to maps or infrastructure, either because they are not prepared or because existing resources are incompatible. Therefore, we envision navigation systems that are capable of operating without depending on such map information.

Removing reliance on digital environmental information (*i.e.*, infrastructures and prebuilt maps) introduces several technical challenges. First, without access to such

information, the system cannot determine which direction to take to reach the destination. Second, unlike conventional navigation systems that rely on prebuilt maps to decide when and what information to convey, a map-less system must detect environmental features through onboard sensors and determine which of the detected information is important to convey to the user, despite lacking predefined priorities. Taken together, since the system alone cannot fully determine the route and information to convey, it is essential to develop methods that enable both the user and the system to collaboratively identify the path to the destination, while also designing approaches that can convey necessary information from onboard sensors.

To overcome this challenge, we adopt collaborative interaction between humans and the system in which the user acts as the system's sensor or decision-making process [20]. We aim to use this interaction to allow users to complement missing information that comes from a map-less setting. For example, Fallah *et al.* [20] showed one collaborative interaction where the users act as a systems sensor in which users confirm if the navigation command (*i.e.*, "go straight until you reach hallway intersection") has been successfully executed. This allows the users to help the system's localization process even without infrastructures, which benefits blind users by enabling the system to convey accurate directional cues. Another example of collaborative interactions can be seen in the relationship observed between guide dogs and blind users, in which guide dogs support the user's mobility while the user determines the direction [21]. For example, a blind user may instruct the dog to move forward, and then the dog walks straight until it finds an intersection. It then orients its head toward the possible directions of travel and conveys this information through the harness. The user gives the next command based on the directional cues perceived through the harness. By repeating this interaction, in which users are involved in the decision-making process, they can reach the destination. Still, the abovementioned use case is typically constrained to situations where users are already familiar with the route and can determine the way independently. Therefore, we extend this collaborative interaction so that it remains effective even in unfamiliar environments by incorporating methods that convey information beyond what guide dogs can provide, such as signboards and other environmental cues. In short, inspired by existing collaborative interactions in which users actively participate in the system's sensing and decision-making processes, we aim to address the challenges of the map-less setting, where the system lacks environmental information.

In this thesis, towards the realization of map-less navigation systems, we aim to answer the research question:

RQ1 How can we adopt collaborative interactions with blind users for map-less navigation systems?

In addition, it is essential to examine how map-less navigation systems compare with existing aids (*e.g.*, a cane). Since map-less systems have access to less environmental information than map-based systems, they may demonstrate reduced performance. Therefore, we aim to answer the question:

RQ2 How do map-less navigation systems compare with existing aids or map-based systems?

Even with collaborative interaction, route information to the destination, as well as information conveyed during the navigation, remains necessary to enable users or systems to determine their way based on it. Therefore, this raises another research question:

RQ3 What kind of route information sources and information conveyed during the navigation can be used to complement map-less navigation systems?

To this end, we have developed two smartphone systems called Corridor-Walker [22] and Snap&Nav [23], two robotic systems called PathFinder [24] and WanderGuide [25], and one benchmark called Memory-Maze [26]. Through the design, development, and evaluation of these systems and benchmarks, we aim to answer these research questions for realizing map-less navigation systems.

TABLE 1.1: **Chapter Overview.** We first describe two smartphone-based map-less navigation systems: one adopts a relatively simple scenario in which users are familiar with the building, and the other captures a floor map image as an information source for the environment. Both smartphone systems share a common collaborative interaction, which is scanning. We then describe two map-less navigation systems that use navigation robots as platforms to investigate how map-less navigation can assist blind users with higher-end devices. The first robot system addresses navigation by adopting a guide-dog-like collaborative interaction, and the second addresses an exploration scenario. Finally, we describe a VLN benchmark that targets map-less navigation scenarios.

System/Benchmark Name	Device	Core Technology	Scenario
Corridor-Walker [22] (Chapter 3)	Smartphone	Intersection Detection Route Planning	A blind user knows the route to the destination, but the environment has many obstacles and intersections
Snap&Nav [23] (Chapter 4)	Smartphone	Intersection Detection Floor Map Analysis	A blind user is unfamiliar with the environment and sighted assistance help user capture floor map
PathFinder [24] (Chapter 5)	Robot	Intersection Detection Sign Recognition	A blind user is unfamiliar with the environment and navigate based on the route description by sighted passersby
WanderGuide [25] (Chapter 6)	Robot	Waypoint Detection MLLM-Based Scene Description	A blind user is unfamiliar with and explores the environment such as shopping mall or museum
Memory-Maze Benchmark [26] (Chapter 7)	Simulator/Robot	Visual language navigation	A robot autonomously navigate based on the route description by sighted passersby

1.2 Dissertion Organization

Table 1.1 shows the overview of the chapters. To develop map-less navigation systems and to answer the research questions, this dissertation presents five research projects. Specifically, we present systems that use smartphones [22, 23] and navigation robots [24, 25] as a platform, respectively. Beyond work in the human-computer interaction (HCI) field, this dissertation developed a visual language navigation benchmark [26], which is a technology that can be applied to a map-less navigation system. We began with a simple setting using minimal hardware and gradually progressed to more complex scenarios and devices. Below, we summarize the aim and abstract of each chapter

1.2.1 Chapter 2 Literature Review

We first summarize the differences between the five research projects presented in this dissertation and the related literature, situating this dissertation within the broader line of prior work. We begin by describing the challenges faced by blind people and the current state of available solutions. We then review existing map-based navigation systems, followed by a discussion of how obstacle avoidance systems, which sometimes operate without maps, serve a fundamentally different purpose. We further extend this survey to the robotics domain. Next, we explain

the control methods adopted in assistive navigation as well as those found in the broader robotics and HCI literature. Finally, we situate the proposed work within the existing visual language navigation (VLN) task, which can be incorporated for map-less navigation systems.

1.2.2 Chapter 3 Corridor-Walker: Mobile Indoor Walking Assistance for Blind People to Avoid Obstacles and Recognize Intersections

We first investigate the collaborative interaction in map-less navigation system by using an off-the-shelf smartphone, a device widely used among blind people. As a first step, we adopted a simple scenario where the user knows the route to the destination inside a building, either through prior experience or by asking people around them. This study developed a system called *Corridor-Walker* [22], which relies solely on real-time sensing to detect and convey obstacles and intersections. Users engage in collaborative interaction by scanning the environment, which allows the system to accumulate additional environmental information to better assist in map-less navigation scenarios. A user study showed that scanning interaction helps users better perceive intersection shape, improving environmental understanding and enabling safer navigation to destinations. However, the system exhibited slower task completion times compared to using a white cane.

1.2.3 Chapter 4 Snap&Nav: Smartphone-based Indoor Navigation System For Blind People via Floor Map Analysis and Intersection Detection

Building on Corridor-Walker, which assumed that users are familiar with the route, this research developed *Snap&Nav* [23], which incorporates floor map recognition to obtain environmental information as an information source to the destination. Here, we considered scenarios where the user is completely unfamiliar with the building and asks a nearby sighted assistant to take a photo of the floor map. Using this image, the system then navigates the user to their destination. This study focused on designing a preliminary floor-map recognition algorithm and a tracking method using intersection detection, and it investigated whether collaboration with sighted people is acceptable and feasible for blind users. Through user studies involving both sighted and blind participants, we found that each group successfully accomplished their respective tasks: sighted participants captured floor map images, while blind participants navigated to the destination. Both groups accepted the system design, including the integration of a sighted assistant within the workflow. The research also highlighted the need for improved floor map recognition algorithms capable of analyzing complex and varied floor maps, as well as intersection detection algorithms that can accurately identify intersections of diverse shapes.

1.2.4 Chapter 5 PathFinder: Designing a Map-less Navigation System for Blind People in Unfamiliar Buildings

To expand the system's assistance capability, this research transitioned to a robotic platform equipped with richer sensors, greater computational power, and autonomous mobility, resulting in the development of *PathFinder* [24]. The first two studies relied on smartphones, but their sensing and computing capabilities were limited, for example, LiDAR sensors with a restricted range. In addition, users were required to

manage mobility themselves, which could hinder decision-making, such as determining which intersection to approach and which direction to turn. By adopting a robotic system, intersections of various shapes can be detected more reliably, user mobility can be taken over by the system, and decision-making capability can be enhanced. PathFinder addresses scenarios in which a blind user navigates an unfamiliar building using route descriptions provided by sighted passersby. The robot adopts a guide-dog-like interaction model: the robot takes over mobility, while the user takes charge of orientation. This research also revealed information requirements in the abovementioned situation, such as the need for sign information, and essential functionalities, such as the ability to return to the initial location, through a participatory study. Based on these investigations, we improved PathFinder and conducted a user study, which revealed that blind users were able to determine their route in unfamiliar buildings using PathFinder.

1.2.5 Chapter 6 WanderGuide: Indoor Map-less Robotic Guide for Exploration by Blind People

Given our future vision where map-less systems are used in various new places, it is important to assist exploratory tasks, such as window shopping or freely walking around a floor to learn its layout. Therefore, this chapter introduces *WanderGuide* [25], a map-less exploration system for blind people. Using multimodal large language models (MLLM) capable of describing diverse environmental information and an additional map-less navigation algorithm that allows the robot to navigate in open space, which can be seen in shopping malls, this research extended PathFinder to exploratory navigation. Through a formative study conducted at a shopping mall and a science museum, this work revealed requirements for WanderGuide, such as the need to provide more concrete information, the ability to customize the description length generated by the MLLM, and the functionality to return to previously explored locations. Based on these investigations, we improved WanderGuide and conducted a user study, which revealed that blind users were able to enjoy exploration and find the locations they found interesting.

1.2.6 Chapter 7 Memory-Maze: Scenario Driven Visual Language Navigation Benchmark for Guiding Blind People

The last project developed a visual language navigation (VLN) benchmark called *Memory-Maze* [26]. VLN is a task in which an agent (*i.e.*, in our case, a robot) uses visual perception through an RGB camera to determine its path to a destination, based on a provided route description. This scenario is the same as that addressed in the PathFinder study: blind users receiving verbal navigation instructions from sighted passersby in unfamiliar buildings. Therefore, VLN models have the potential to automate the navigation process for PathFinder. However, existing VLN research had not addressed instructions spoken from sighted passersby based on their memory. To address this gap, Memory-Maze was developed. It mimics real-world instruction characteristics and supports future technical development. Through an experiment with existing state-of-the-art models, we demonstrated that current VLN models fail to adequately address this scenario, highlighting the need for further research and development.

1.2.7 Chapter 8 Discussion and Conclusion

Finally, in this chapter, we summarize the overall findings. Although navigation with our systems required more time, they generally increased user confidence and reduced cognitive load compared to regular aids. Users also accepted the systems and recognized their potential for application in various environments. A key factor in this outcome was the adoption of collaborative interaction, such as scanning or decision-making. We further discuss the benefits of this approach, particularly its ability to support users in completing the task of perceiving the environment more effectively. We also revisit the research question presented above and outline open future directions, such as enhancing shared control dynamics to further improve our system.

Chapter 2

Literature Review

This chapter aims to situate the topic of map-less navigation systems for blind people within a broad line of work.

2.1 Difficulty Faced When Navigating and Exploring

This section describes the general difficulties faced by blind people and their need for assistance when navigating familiar and unfamiliar buildings, as well as when exploring unfamiliar environments, scenarios that we address in this dissertation.

2.1.1 Navigation Challenges

Navigating independently is difficult for blind people, even in familiar indoor environments. They are typically trained in orientation and mobility (O&M) skills by specialists to support independent travel [2]. For example, they commonly use techniques such as trailing with a cane and walking along walls to maintain orientation [27]. The cane is also used to detect obstacles; however, various objects may still be placed along the walls, including elevated obstacles that are difficult to detect [28]. In addition to avoiding obstacles, blind people must understand the geometric structure of the environment [29, 30], such as recognizing intersections, in order to reach their destinations. Even in a familiar environment, blind people might miss an intersection or turn into the wrong one, which may cause disorientation and lead to getting lost.

To navigate correctly, they must reliably perceive the position and shape of each intersection they encounter, while avoiding obstacles. Even with a cane, blind people may collide with obstacles, which can result in damage to both the user and the object. Moreover, the limited range of contact offered by a cane means that intersections may not always be detected, leading to situations where a user unknowingly walks past one [31]. Guide dogs offer an alternative means of support by helping users avoid obstacles and locate intersections [31]. However, not all blind people prefer guide dogs, as they require ongoing care and responsibility [32, 33]. Additionally, guide dog availability is limited, for example, in the United Kingdom, only about 5,000 guide dogs serve approximately 360,000 legally blind individuals, roughly 1.4% of the population [33]. Therefore, in Chapter 3, we first develop a smartphone-based system for blind people to safely navigate by avoiding obstacles and detecting intersections. In this system, we assume that blind users are familiar with the environment and focus on algorithm development. Based on the system and the findings obtained, we then proceed to more challenging scenarios, such as unfamiliar environments, as described next.

2.1.2 Navigating in Unfamiliar Blindings

Navigation in unfamiliar buildings poses significant challenges. Determining one's current location, identifying a route to the destination, and maintaining orientation are all difficult without visual information or sufficient knowledge of the environment [28]. Nonetheless, Engel *et al.* [4] report that 59.4% of 63 blind participants travel to unfamiliar buildings several times a week despite these difficulties. There are two common strategies blind people use to find a route to a destination [4]. One approach is to search for textual route descriptions on the internet. Although this may seem feasible, such descriptions are often unavailable, and preparing them can be time-consuming. As a result, some blind people do not prepare a route at all. The other approach is to ask sighted people on-site for a route description, which requires no prior preparation. Still, these methods do not guarantee that blind users will reach the destination, making the use of a sighted companion one of the most common options [4, 28, 34]. Nonetheless, many blind people prefer to navigate independently without relying on sighted assistance, which is an important motivation for our research [4].

To address this problem, in this dissertation, we tackle independent navigation in unfamiliar environments in Chapter 4 and Chapter 5. A common aspect shared by these systems is that these systems externally source route information to the destination. In Chapter 4, this information is obtained from a floor map of the building, typically placed on the entrance wall. A sighted assistant captures this image once, realizing a form of collaboration between the system and sighted people. The system then analyzes the map, allowing the blind user to remain independent thereafter. In Chapter 5, route information is obtained from descriptions provided by sighted passersby, which reflects a typical situation as described above. PathFinder is designed so that users can reach their destination with greater confidence.

2.1.3 Exploring Unfamilair Blindings

Exploration is also important for blind people, both for familiarizing themselves with unfamiliar environments [35] and for enjoying that place for recreational purposes (*e.g.*, museums [18] or shopping malls [36]). Although learning routes and points of interest (POIs) in a building can sometimes be done by searching online [4], on-site exploration remains valuable because it provides rich sensory information and fosters a stronger sense of independence [36], which in turn motivates blind people to explore on their own [37, 36]. However, independent exploration is challenging, as blind people typically require a sighted companion who can explain visual information during the activity. While some exploration can be supported by existing navigation systems [9], safety concerns often dominate the cognitive load and limit the quality of the exploratory experience. Therefore, in Chapter 6, we propose WanderGuide, a robotic system that assists blind people in exploring without maps. By leveraging the robot's autonomous navigation capability, coupled with recent MLLM advancements in describing the surrounding environment, we aim to design and develop this system.

2.2 Assistance Systems for Blind People

This section describes map-based navigation systems and outlines the main motivation of this dissertation. We then explain how our systems differ from existing

obstacle avoidance approaches, which are map-free but serve a fundamentally different purpose.

2.2.1 Navigation Systems Based on Maps

Most popular approaches for navigation assistance systems for blind people rely on digital prebuilt maps that contain static route information and localization infrastructures. These systems typically take the user's destination as input, plan a path on the route map, and provide turn-by-turn guidance. Such systems include those developed by industry (*e.g.*, Google Maps [38] and BlindSquare [12]) using global positioning systems (GPS), as well as those developed in academia using magnetic information [39, 40], visual features [10, 8, 41], radio frequency identifier (RFID) tags [42, 43, 44, 20], visible light communication (VLC) [45], and BLE beacons[9, 46, 47, 48]. By referring to the digital prebuilt maps, those systems could also provide descriptions of surrounding POIs, sometimes even by their priority level [9, 13]. These systems have been proposed in various devices, such as smartphones [49], handheld haptic devices [50, 51, 52], wearable devices [10, 53, 54], and robots [55]. Each type of device offers unique advantages - Smartphones and handheld haptic devices are portable; Smartphones are also widely used by blind people [56, 57]; Wearable devices free the user's hands [54]; And robots are able to autonomously guide users [11]. Among them, robots generally take two forms: quadruped and wheeled. While wheeled robots are unable to navigate stairs in the same manner as quadruped robots [16], with current quadruped robots, blind users often find wheeled robots more suitable due to their silence and stability [58]. Among wheeled robots, suitcase-shaped robots [17, 59, 60, 18, 61] have become particularly practical and are therefore widely used in research. The appearance of suitcase-shaped robots allows blind users to blend seamlessly into their environment, leading to higher social acceptance from users, surrounding pedestrians, and facility managers [60].

However, almost all of the systems mentioned above cannot be used in a new place on the fly, and only a limited number of navigation systems (*e.g.*, InclusiveNavi [62] and BlindSquare [12]) are publicly deployed. This is because prebuilt maps and localization infrastructures are expensive to install and typically require technical experts to manage. Furthermore, maps and infrastructures are costly to maintain. This motivates the development philosophy of *map-less* navigation systems, in which no environmental information is prepared for the system prior to navigation.

2.2.2 Obstacle Avoidance Systems

Obstacle avoidance systems assist blind people in navigating safely by avoiding obstacles (*e.g.*, walls [63, 64], boxes [65, 63, 49], chairs [10, 64], poles [66, 49]) through real-time sensing using modalities such as RGB cameras and LiDAR sensors [67]. These approaches often inform users of the position of obstacles [49, 68, 64, 69, 70, 66, 63, 71], but a limitation is that users themselves must determine their path based on the given feedback. There are also systems that guide users by generating safe paths to avoid obstacles [54, 72, 73, 11, 59, 10]. These include map-based approaches, commonly used in robotic systems, which rely on predetermined maps and detect obstacles from differences between the map and sensed information, as well as map-free approaches that operate solely on real-time perception. While these map-free obstacle avoidance systems may seem similar to map-less navigation, they fundamentally differ in purpose. Map-less navigation for blind people aims to support global tasks such as navigating to a specific destination or exploring an environment to learn and

move toward areas of interest—tasks that require handling orientation and broader spatial decision-making. For example, our system Corridor-Walker (Chapter 3) emphasizes intersection detection to correctly navigate users to their destination, while also performing obstacle avoidance. In contrast, existing obstacle avoidance systems primarily ensure local safety while leaving orientation and overall navigation decisions to the blind user.

2.3 Robotic Systems without Maps

Robotics has examined various algorithms that either rely on maps or operate without them. A simple story describing these scenarios is presented by Guerrero *et al.* [74], who outline a case in which a navigator must return from one place to another through a long and unfamiliar route. First, the navigator could obtain a map of the building to retrace the path, representing a map-based approach. However, as discussed above, such a map may not always be available. Second, the navigator could memorize the sequence of steps and turns taken. This corresponds to a dead reckoning approach, in which the agent estimates its position relative to its starting point using accumulated motion information. Third, the navigator could determine the return path by observing environmental cues such as signs. This represents a map-less approach. Finally, the navigator could build a map while exploring and use it later to navigate back, which is a map-building approach. In this section, we focus on map-less and map-building approaches, as these are most relevant to this dissertation.

2.3.1 Map-less Approach

Map-less approaches in robotics use no maps created before navigation, and their goal is typically to guide a robot to a destination or accomplish a task. For navigation tasks, instead of relying on prebuilt maps, these systems utilize alternative forms of information to move toward the target location. These approaches can be classified into four categories [75, 76, 77]. *Optical flow–based approaches* match real-time RGB images with sequences for path following [78]. *Appearance-based approaches* train on environmental information (*e.g.*, images) and associate it with localization cues or control commands to learn an environmental representation (without constructing a map) [79]. *Object recognition–based approaches* store detected objects in a 2D grid representation and generate plans using symbolic commands, such as language instructions like "go to the nearest blue sofa." [76] *Feature-based methods* use image sequences to extract relative displacement to the destination [80]. Other approaches include using an image of the target location combined with reinforcement learning to learn a policy that navigates to the goal [81, 82]. While this dissertation uses the term "map-less" in the same sense as robotics literature, we explicitly clarify that, within the scope of this work, maps are not prepared prior to navigation but may be created dynamically during navigation and reused. Examples include PathFinder (Chapter 5) and WanderGuide (Chapter 6). This approach is also similar to the map-building method described next.

2.3.2 Map-Building Approach

Map-building approaches in robotics have no map prior to navigation but instead create maps during navigation so they can be used later (*e.g.*, for underground exploration). The most representative approach is Simultaneous Localization and Mapping (SLAM), which constructs a map by extracting landmarks, associating sensor data, estimating the robot's state, and continuously updating both the state and the landmark positions [80]. Another line of examples of these approaches includes the undermine navigation and construction of topological maps by detecting intersections [83] and occupancy maps using RGB images and deep reinforcement learning [84] while the robot is navigating the space.

In summary, map-less approaches in robotics aim to complete navigation tasks using real-time sensing, whereas map-building approaches focus on constructing maps during navigation for later use. In this work, to assist blind people, our system lies at the intersection of these two approches: the real-time map constructed during navigation directly contributes to wayfinding at the very same moment it is being built (Chapter 5 and Chapter 6). Furthermore, the resulting map representation must be not only suitable for robotic decision-making but also understandable and usable for the blind user.

2.4 Collaborative Interaction Methods

In this section, we describe the positioning of our collaborative interaction approach within the HCI field, followed by its positioning within the robotics field. Finally, we situate the approach within the assistive technology domain.

2.4.1 Human-in-the-Loop Interaction in HCI Domain

As navigation systems cannot determine the path to a destination without prebuilt maps or reference information, this dissertation employs a collaborative interaction. In this interaction, the system assists the blind user, but the user also intervenes in the system's operation and takes part in the sensing or decision-making process. By adopting this interaction, the system can provide more effective navigation support. This kind of interaction paradigm is also referred to as Human-in-the-Loop [85]. Notable interaction paradigms include mixed-initiative interaction [86], in which both the system and the user take initiative during an interaction. For example, not only can users ask the assistive system questions when needed, but the system can also ask users questions so that humans can complement the imperfections of artificial intelligence (AI) [87]. In the intersection of AI and accessibility, efforts also include developing AI systems to convey how users should adjust the orientation of a smartphone when an image lacks sufficient features because it is pointed in a slightly incorrect direction [88]. In this sense, the interaction in our domain can be viewed similarly as a form of collaborative interaction that complements the system's limited environmental knowledge in a map-less setting. For example, as explained in the introduction, Fallah *et al.* [20] used blind users as sensors to complement the system's localization process, even in environments without infrastructure. This shows that users can contribute to the system's capabilities not only in the decision-making process but also in the sensing process. Therefore, for smartphone systems, we employ scanning interaction, allowing the user to assist the system in obtaining information and thereby achieving better recognition of the environment (Chapter 3 and Chapter 4). We also incorporate intervention in the decision-making process for

robotic systems, where the user specifies which way or place to go instead of the system (Chapter 5 and Chapter 6).

2.4.2 Shared Control in Robotics Domain

In robotics, collaborative interaction is often refer to as shared control [89], *i.e.*, a method to control a robot using both human decision and the functionality of a system. According to Wang and Zhang [90], shared control is defined as a *"case in which the robot motion is determined by both the human operator and robot decisions in a mostly balanced fashion."* Shared control can be separated into near-operation, in which the operator perceives the scene with their direct sense, and teleoperation, in which the operator perceives the scene indirectly, such as through a screen. For example, near-operation has been used for assisting a driver to keep their vehicles in lane [91], controlling a wheelchair [92, 93, 94], and for assisting blind people to navigate in familiar buildings [95, 96, 97], while teleoperation has been used for navigating where a human cannot go [98, 99, 100], or for reconnaissance [101]. In our robotic systems (Chapter 5 and Chapter 6), we employ near-operation interaction. This interaction is inherent to the design, as the robot guides the user in real time. However, our near-operation interactions enable users to utilize their own sensing capabilities, such as detecting intersection directions or perceiving the spatial atmosphere, which are important for wayfinding and exploration.

2.4.3 Shared Control in Assistive Systems for Map-Free Navigation

The systems presented in this dissertation adopt near operation shared control, as users navigate together with the robot in order to reach destinations or explore. Prior examples include users specifying directions at intersections while the robot provides automated guidance to the next intersection [35, 15, 95, 96], as well as users controlling speed while the robot handles lateral movement [102]. The intersection with robotics has also advanced interactive control methods, such as employing leash and haptic rein mechanisms with quadruped robots [103, 104, 105]. It is also worth noting that the preferred level of control by blind users, such as choosing between shared control or fully autonomous robotic navigation, may vary depending on context [102]. These interactions enable a map-free approach, as the direction is determined by the user rather than the robot. While these shared control systems are relevant to our setting, many prior studies do not yet assume a concrete real-world scenario. For instance, experimenters often tell participants where to go at each step during user evaluation [35, 15, 95, 96]. In this dissertation, we adopt shared control for navigation in both familiar and unfamiliar buildings, as well as for exploration in unfamiliar buildings. We examine the extent to which users can take part in the decision-making process. To support users in decision-making, our robotic systems (Chapter 5 and Chapter 6) convey environmental information, as described in the next section.

2.5 Conveying Environmental Information for Blind People

As map-less navigation requires users to make decisions based on the information conveyed by the system, the design space demands careful consideration. The system cannot simply present all available information; it must communicate only what is necessary and sufficient for informed decision making. Exploration of the design space of assistance systems by investigating information needs for blind people in

familiar or unfamiliar spaces is an ongoing research field [106, 107, 108, 109]. By using visual captioning [110] and question-answering models [111], researchers have developed systems for assisting their scene understanding by conveying information such as traffic lights [112, 113], doors [114, 110], intersections [9, 20, 95], and signs [115, 110, 116, 110]. Alternatively, remote sighted assistance (RSA) applications (*e.g.*, Aira [117] and BeMyEyes [118]) have long been a practical aid for providing blind people with assistance in various tasks, such as scene describing and turn-by-turn guidance. However, RSA system service quality depends heavily on the sighted assistance provided [119] and may also not be sustainable due to the need for an available online assistant.

Additionally, with the emergence of LLMs and MLLMs, scene describing systems (*e.g.*, Seeing AI [120], BeMyAI [121] and GPT4o-demo [122]) have been developed, which enable blind people to understand scenes in diverse scenarios [123, 124]. ChitChatGuide [13] employs LLMs to interpret predetermined maps and deliver exploration-related information during navigation to a specified destination. MLLM-based systems, such as WorldScribe [125], analyze captured images to provide real-time contextual descriptions and can adapt the level of detail based on user context, such as device movement speed.

As information needs are context-dependent [126], system requirements change depending on the scenario we aim to address. Therefore, we adopt a participatory design for information conveying functionalities. For example, in PathFinder (Chapter 5), we reveal that sign information is critical in navigation scenarios where the system relies on descriptions provided by sighted passersby in collaboration with the navigation robot. For WanderGuide (Chapter 6), we reveal various information needs during exploration in science museums and shopping malls, as well as how MLLMs should describe surrounding information to provide a better exploration experience.

2.6 Visual Language Navigation

Visual language navigation (VLN) is a task in the embodied AI field in which an agent (*e.g.*, a navigation robot) with visual access to its surroundings navigates under human instructions [127]. For example, given an instruction such as "turn at the first right and stop when you see a sofa" the robot predicts a sequence of actions that will guide it to the sofa. This task is closely related to map-less navigation approaches in robotics [75]. Early VLN models explored sequence-to-sequence supervised learning [127, 128], where the model predicts a single action (*e.g.*, turn left, turn right, or move forward) based on the current visual input. VLN environments typically include a map representation called a navigation graph, where nodes and their connections are predefined, and the model selects the next navigable node from the current one. VLN research spans various benchmarks, from indoor environments [127, 129, 130] to outdoor settings [131, 132]. However, supervised VLN models are known to generalize poorly to unseen environments [133], as they tend to overfit to the specific environments in which they are trained. With recent advancements in large language models (LLMs), researchers have explored VLN methods that require no retraining [134, 135, 136]. Examples include extracting and following landmarks from instructions [135] and performing iterative node selection in navigation graphs [134]. Because navigation graphs are not available for

all environments, newer approaches generate navigation graphs [137], predict low-level actions directly [138], or produce generative navigation programs that execute low-level robot movements [139, 140].

Despite this progress, there remains a gap between VLN developments and practical use in assistive navigation. For example, the environments used to develop VLN systems do not always reflect those where blind people would realistically use navigation assistance. Iterative inference can increase latency, and many datasets collect instructions in ways that do not align with real-world communication patterns. In Chapter 7, we further describe these gaps and present our approach for adapting VLN technologies to assistive navigation for blind users.

Chapter 3

Corridor-Walker: Mobile Indoor Walking Assistance for Blind People to Avoid Obstacles and Recognize Intersections

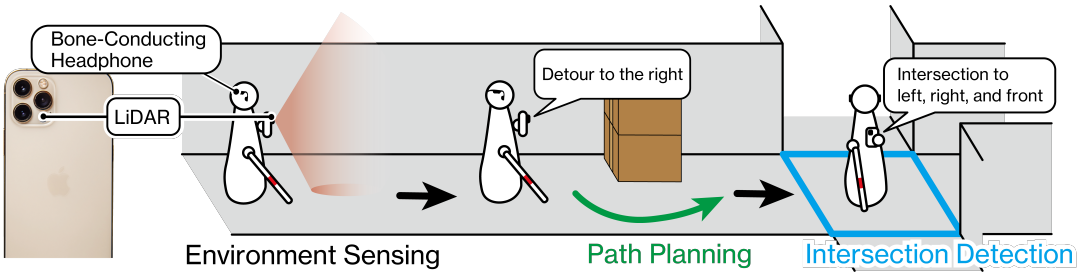


FIGURE 3.1: **Corridor-Walker Recognizes Obstacles and Intersections.** Through audio and haptic feedback, blind users can use the system to detect an upcoming intersection and recognize the paths it leads to. This also prevents them from walking past an intersection unnoticed. Also, they could safely navigate by avoiding obstacles.

3.1 Introduction

We begin by addressing a familiar indoor corridor-like environment, which is simple in structure yet challenging for blind people to navigate. To approach this problem, we consider the use of a smartphone, a device that is already well adopted among blind people [56, 57].

In indoor corridor-like environments, they usually rely on the surrounding walls to navigate [27]. As various obstacles may be placed along the wall, such as wall-mounted furniture and objects [28], blind people may collide with these obstacles, resulting in damage to both the blind people and the objects. In addition to avoiding obstacles, blind people must be aware of the corridor's geometric structure [29, 30], such as intersections, to navigate to their destinations. Walking past an intersection unnoticed or turning into an incorrect intersection could lead to blind people being lost. To navigate correctly, they need to reliably perceive the position and shape of each intersection that they go through.

With the use of only a white cane, they may not be able to locate an intersection, resulting in walking past one unnoticed [31] (Section 3.6.1, A3.1). In addition, the white cane does not fully support the shape recognition of intersections (Section 3.6.3, A3.7) because it has a limited range of contact. Although guide dogs can

help blind people locate an intersection [31], not all blind people prefer them as they require certain caretaking [32, 33]. In this regard, indoor turn-by-turn navigation systems have been proposed to ensure safety by conveying obstacle-avoiding path [59, 141, 10, 54, 64] or positions of obstacles [49, 68, 69], and without being lost via and conveying correct information about intersections [9, 20]. However, such systems require static route maps and additional infrastructure, making them unlikely to be used in various places.

Therefore, we present *Corridor-Walker*, a mobile indoor map-less walking assistance system for supporting blind people in avoiding obstacles and recognizing (*i.e.*, locating and grasping the paths they lead to) intersections (Figure 3.1). The system is aimed to be used in indoor corridors where static route maps and infrastructure are not available, but the user has the knowledge of the turns they need to make to reach the destination (*e.g.*, corridors in apartments, offices, or hospitals) from a prior visit or from tactile maps [142, 143] or interactive devices [144]. The system assists the user in avoiding obstacles by navigating the user to trace an obstacle-avoiding path using both spatialized audio and text-to-speech (TTS) feedback. By detecting intersections, the system informs the user of the existence and shape of an upcoming intersection through vibration and TTS feedback. Notably, the system prompts the user to scan their surroundings to clarify the grid map's layout. Once the shape is confirmed, it informs the user, enhancing system performance and user understanding, realizing collaborative interaction.

To achieve these functionalities, the system first constructs a 2D occupancy grid map [54, 72, 73, 70] of the surrounding environment using a LiDAR sensor equipped with an iPhone 12 Pro [145], which supports accurate grid map construction. Then, the system plans an obstacle-avoiding path on the grid map using the A* path planning algorithm [146]. Simultaneously, the system detects upcoming intersections using the YOLOv3 detector [147]. Since only real-time sensing results are used, these functionalities can be accomplished without the need for a static route map or additional infrastructure.

We conducted a user study with 14 blind participants. The aim of the study was to investigate whether users could employ collaborative interaction, scanning, to identify intersections, and whether the system could be used to support an indoor corridor navigation scenario. The participants were asked to perform three tasks. In the first task, the participants turned in different types of intersections, and were asked to list all directions to which each intersection led. In the second task, several obstacles were placed in a straight corridor, and the participants were asked to walk through it, while avoiding the obstacles. In the last task, the participants were asked to navigate a corridor containing both obstacles and intersections. The study revealed that Corridor-Walker enabled the participants to avoid obstacles while relying less on the wall and to better grasp the intersection's shapes.

3.2 Related Work

In this section, in addition to the related works reviewed in Chapter 2 - navigation challenges (Section 2.1.1), map-based navigation assistance systems (Section 2.2.1), and obstacle avoidance systems (Section 2.2.2) — we describe indoor intersection detection technology and interaction methods applicable to mobile devices.

3.2.1 Intersection Detection in Indoor Environments

The problem of detecting indoor intersections without a static route map has been explored by using LiDAR or RGB sensors. Lacey and Shane proposed the use of a Bayesian network to detect intersections using a 180° laser range finder attached to a robot [95]. Garcia *et al.*proposed detecting intersections from RGB images taken by a quadcopter using a rule-based approach [148] and a convolutional neural network [149]. These methods can adequately detect various types of intersections in indoor environments in advance. However, applying these methods to navigate blind people may not be suitable as it has been shown that some blind people may not hold smartphones stably [8, 150], and therefore images captured by them may contain motion blur [151]. In addition, images captured by them may miss the entire subject from the camera [152]. To overcome these issues, our approach uses an image of a 2D occupancy grid map of the surrounding environment, which is constructed using a LiDAR sensor equipped on a smartphone and is thus less susceptible to motion blurs. Moreover, we included collaborative interactive feedback that can guide the users to scan the environment when more information is needed to identify intersection types.

3.2.2 Designing Non-visual Feedback for Blind People

Based on previous studies, we designed the Corridor-Walker interface to provide multiple feedback modes through bone-conducting headphones, with each mode conveying specific information suited to different situations. To convey navigation information to a blind user, previous studies utilized feedback using either audio feedback (*e.g.*, TTS [8, 153, 9, 20, 10, 114, 154, 116], sonification [155, 49, 41, 114, 68], spatialized audio [156, 120, 157, 158, 68, 159, 66], beep sounds [61, 59, 66, 66]), vibration feedback [8, 150, 63, 64, 69], or thermotactile feedback [160, 161]. Although instructions from TTS are capable of conveying various clear instructions to users, their use should be kept minimal. This is because they may block ambient sounds that blind people often rely on [162], may not be heard in a noisy area [163], and may harm their cognitive load [164]. In addition, although TTS can convey various instructions, it is a challenge for them to use TTS for slight adjustments of their orientation [9] (*e.g.*, rotate 4° to the right). In contrast, according to Lock *et al.* [157], slight adjustment of a user's orientation with spatialized audio using bone-conducting headphones was found to be effective. In addition to audio feedback, vibration feedback is used to convey simple instructions to blind users. Blind people highly prefer them as they can be perceived in noisy environments [59] and do not harm the person's cognitive load compared to audio feedback [164]. Although Nasser *et al.*reported that thermotactile feedback outperforms vibration feedback in providing directional cues [161], it may require additional Peltier modules with smartphones.

3.3 System Design

Our main goal is to support the blind user in navigating indoor corridors to safely arrive at their destination without modifying the infrastructure of the building or requiring its static route map. The typical situation is as follows: *A blind person is walking in an indoor corridor, such as those in offices, hospitals, and hotels. The person knows how many intersections they have to turn to reach the destination. However, there are several obstacles in the corridor, which are blocking the path.* The aim of the system is

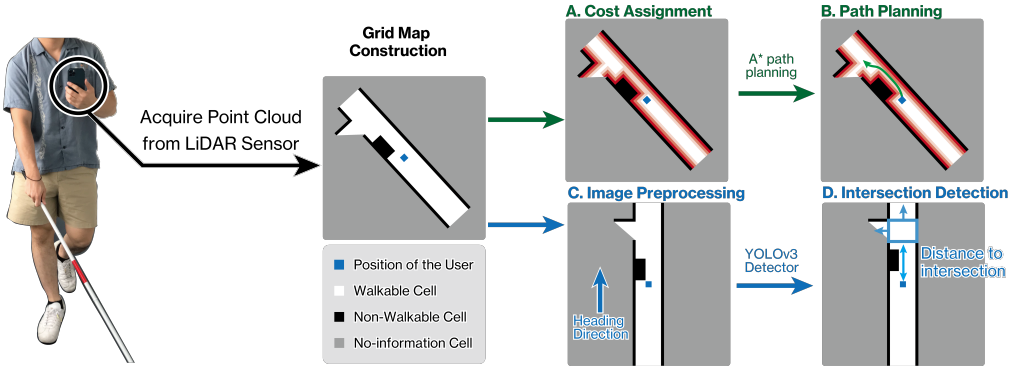


FIGURE 3.2: **Techniacl Overview of Corridor-Walker.** First, the system constructs a 2D occupancy grid map from the point cloud acquired from the LiDAR sensor. A) Then, the system assigns a cost value to each cell, and B) plans an obstacle-avoiding path. Simultaneously, the system C) preprocesses the image of the grid map, and D) detects upcoming intersections through the YOLOv3 detector.

to augment the function of a white cane to help users avoid obstacles and recognize intersections.

3.3.1 Avoiding Obstacles

Blind people often walk along walls when navigating indoor environments [27]. Simultaneously, many obstacles are usually placed along the wall [28]. This may result in accidents where they collide with obstacles. Therefore, we aim to guide them along a path that avoids such obstacles. We designed the system that can generate a path that keeps a distance from nearby walls and obstacles, and navigates the user to trace the path without veering. In other words, the system assists the user in walking without relying on the wall, preventing collisions with obstacles placed along the wall. If there are obstacles ahead, the system generates a path that circumnavigates the obstacles and guides the user to make a detour around them.

3.3.2 Detecting Intersections

To navigate to a destination, blind people need to perceive the positions and shapes of intersections they have to go through. However, in situations where they cannot walk along a wall (*e.g.*, there are obstacles along the wall), they may walk past an intersection without noticing [31]. In addition, neither white canes nor guide dogs support recognition of the intersection shape. To augment the ability of traditional navigation aids, we designed our system to inform users of an upcoming intersection and its shape. The system will notify the user of possible existence of an intersection ahead to prevent them from walking past it. Then, information about the shape of the intersection is provided to the user once they reach it.

3.4 Implementation

In this section, we present the details of the implementation of the proposed system. Figure 3.2 shows an overview of Corridor-Walker.

3.4.1 Device and Ergonomics

Corridor-Walker was implemented on an off-the-shelf smartphone, the iPhone 12 Pro [145]. This is because the smartphone is equipped with a LiDAR sensor with a maximum sensing range of 5 m [165] and can be obtained at 60 Hz. Also, since the LiDAR sensor emits infrared lasers to measure the distance between the sensor and objects, the system can work well in an environment without strong sunlight, or even under low-lighting conditions. It is equipped with a 16 Core Neural Engine, which could efficiently run machine learning models with the CoreML toolkit[1]. The user will be asked to hold the smartphone in front to scan the environment, as illustrated on the left side of the Figure 3.2.

3.4.2 Grid Map Construction

The 2D occupancy grid map is constructed through a point cloud acquired from a LiDAR sensor and by using the localization algorithm provided by the augmented reality kit (ARKit) [166]. Therefore, the grid map will contain the grids that are in front of the user and the grids of the path through which the user walked. To determine whether each grid is walkable or not, the normal vector of each point is calculated [167], and the system determines floor plane using the random sample consensus (RANSAC) algorithm [168]. To operate the algorithm in real-time, the point cloud and normal vector computations are performed using the Metal toolkit[2], which enables parallel processing on the GPU. If a point has a normal vector that is parallel to the gravity vector and its height is within 0.1 m from the height of the floor plane, it is classified as a point belonging to the walkable area. Other points are considered as points belonging to the non-walkable area. Then, the cell in the xy-plane grid map on which each point is projected is determined. Each side of a cell is 0.15 m long on a real-world scale. When a cell contains more points that belong to the walkable area than those of the non-walkable area, the cell is labeled as a walkable cell (white pixel in Figure 3.2). Otherwise, the cell is labeled as a non-walkable cell (black pixel). If a cell contains no points, it is labeled as a no-information cell (gray pixel). Moreover, the system updates the label of each cell each time it is observed. This allows the system to handle dynamic obstacles, *e.g.*, other pedestrians and cleaning robots. This is because the system will initially label the cells occupied by obstacles as non-walkable cells, and once those obstacles move away, the system will update those cells into walkable cells. While the whole algorithm could run at approximately 50 Hz, we adjusted the frequency to 10 Hz, which opened additional computation space and enabled the system to recognize both static and dynamic objects as non-walkable areas.

3.4.3 Path Planning and Obstacle Avoidance

The system uses a path planning algorithm to guide the user on a safe path. To generate such paths, we utilized the methods commonly used in the field of robotics [169, 170, 171]. We first assign a cost value to each cell in the grid map and then use a path planning algorithm to generate a safe path. In the following section, we describe these steps in detail, followed by their use to avoid obstacles and prevent veering.

[1] https://developer.apple.com/machine-learning/core-ml/
[2] https://developer.apple.com/metal/

Cost Assignment

First, the system assigns a cost value between 0 and β to each walkable cell (Figure 3.2–A). This allows the system to obtain a cost map, *i.e.*, a grid map where each cell is assigned a cost value, which can be used to plan a path far from non-walkable cells to guide the user. To compute the cost for each walkable cell, let δ_i denote the distance from a walkable cell i to its closest non-walkable cell. The cost value of the walkable cell i is given by $\text{cost}_i = \beta(1 - \frac{\delta_i - 1}{\alpha})$, if $1 \leq \delta_i \leq \alpha$, or $\text{cost}_i = 0$, if $\delta_i > \alpha$, where α upperbounds the distance for a walkable cell to have a positive cost. Here, walkable cells that are closer to a non-walkable cell will have higher costs than those which are further away. In Figure 3.2–A, walkable cells with high costs are indicated in dark red, and those with low costs are indicated in light red. Based on our observations, we set $\alpha = 3$ and $\beta = 50$.

Path Planning Algorithm

First, the system searches for the destination to perform a path planning algorithm. To do so, the system samples all walkable cells with the lowest cost at a distance of γ m ahead in a circular sector with a range of 100° forward. Then, the system sets the mid-point of the longest continuous space of the sampled points as the destination. If the calculated destination falls into a non-walkable cell (*e.g.*, pillars or boxes), γ is shortened by 0.5 m and the process is repeated until the destination is found or γ is set to 0 m. Finally, the system calculates the path to the point using the A* path planning algorithm [146, 169] (Figure 3.2–B). As a result, due to the construction of the cost map, the system generates a path that keeps a distance between every obstacle and wall. The path is updated every time the user walks half of the previously planned path. Based on our observation, we initially set the $\gamma = 3.5$m, which is the distance robustly scanned by the LiDAR sensor on the smartphone.

Obstacle Detection

Although the system can plan an obstacle-avoiding path, it is still necessary to notify the user of obstacles to explicitly alert the user to make a detour. To determine whether a non-walkable cell belongs to a wall or obstacle, the system performs plane detection using the RANSAC algorithm [168] on the 2D occupancy grid map. All the cells in the planes detected by RANSAC are determined as walls, and the remaining cells are determined as obstacles. Then, we consider all cells in the circular sector with a radius of 2 m and a central angle of 30° in front of the user to determine if there is an obstacle ahead. If the number of obstacle cells in the circular sector exceeds 30%, the system determines that there is an obstacle ahead and notifies the user (Section 3.4.5).

Veering Detection

To prevent the user from veering off the generated path, the system determines whether the user is facing the correct direction or not (Figure 3.5–Veering). First, the system calculates its orientation by using the localization algorithm provided by ARKit on the grid map. Then, the system calculates the angle θ between the system's orientation and the direction on the path that the user is expected to move to. If the angle θ is larger than 10°, it is determined as the user veering off the path, and the system will notify the user (Section 3.4.5). Otherwise, it is determined as the user staying on the path.

3.4.4 Intersection Detection

The system detects an upcoming intersection using a YOLOv3 object detector [147]. We used the YOLOv3 detector as it runs at around 70 frames per second on iPhone 12 Pro's Neural Engine, combined with the CoreML toolkit. For the input, we used an image from the 2D occupancy grid map. The position of the generated bounding box (blue rectangle in Figure 3.1) represents the position of the intersection in the real world, and its label identifies the shape of the intersection. As the system uses a grid map constructed from the LiDAR sensor, it is not affected by motion blur, which may occur when blind users take photos using RGB cameras [151]. Therefore, the system can detect upcoming intersections robustly.

Image Preprocessing

As the grid map itself does not contain information about the direction the user is heading, we preprocess the image of the grid map such that this information becomes apparent. Thus, the system rotates the image of the grid map so that the heading direction of the user faces up (Figure 3.2–C). The heading direction of the user is calculated according to their position over the last four seconds. Then, we shift the image so that the user's position is at the center of the image. This preprocessed image (128×128 pixels) is used as the input to the YOLOv3 detector.

Training the YOLOv3 Detector

We trained the YOLOv3 detector to detect upcoming intersections. To train the YOLOv3 detector, we collected 9940 preprocessed images from the corridors of Waseda University. As shown in the Figure 3.3, we annotated the locations of intersections and their shape labels (*i.e.*, the directions it leads to). For example, an intersection that leads only to the left will be labelled as "*Left, Back*" as it leads to the left and the back of the user. Since the intersections with the labels of "Left, Back" or "Right, Back" have the same topological shape, they are defined as "*L-Shaped*" intersections. Similarly, other intersections are classified as "*T-Shaped*," "*Rotated T-Shaped*," and "*X-Shaped*". We set the confidence threshold of the YOLOv3 detector to 0.2, which is based on our empirical observation that this value provides early detection of upcoming intersections with good accuracy.

Determining the Distance to Intersection

The distance between the user and the detected intersection is defined by the number of pixels between the bottom side of the bounding box and the center of the image. The blue arrow in Figure 3.2–D shows an example of the distance between the user and the intersection. As each pixel (*i.e.*, cell) represents 0.15 m in the real world, the number of pixels multiplied by 0.15 m is the distance to the intersection. When the generated bounding box includes the center of the image, it means that the user is at the detected intersection.

Evaluation

To evaluate our detector, we constructed a dataset consisting of 1215 preprocessed images taken in a different location from the training dataset. We measured the following metrics: (1) precision and recall at different distances, and (2) the furthest distance to detect each intersection shape. For the first metric, we measured the

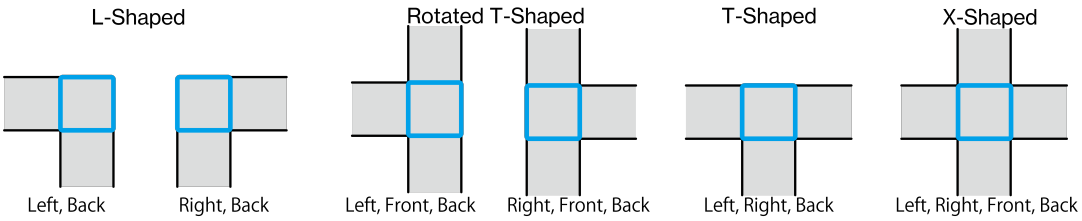


FIGURE 3.3: **Annotation of Intersections.** The labels of the annotated intersections contain all the navigable directions. There are nine lables in total, and six are shown.

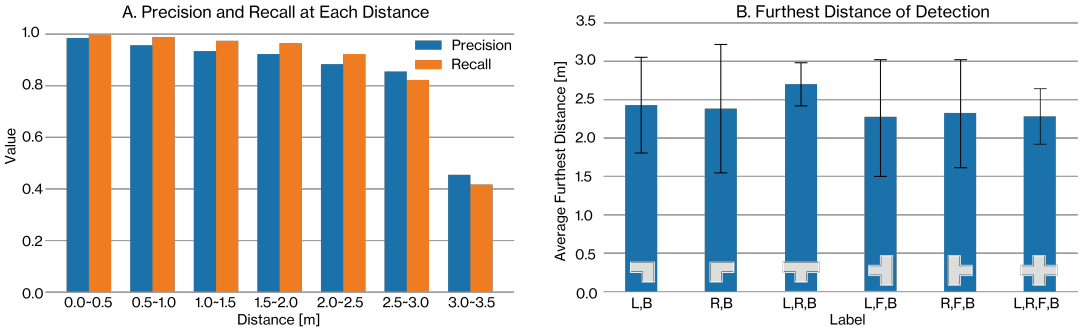


FIGURE 3.4: **Intersection Detection Evaluation.** A) Bar graph of precision and recall at each distance. B) Bar graph of the furthest distance of detection for each label of intersection.

precision and recall of the intersection detection at every 0.5 m interval. Figure 3.4–A shows the results for the first metric. The precision and recall are high when the distance between the intersections and the user is small, but they decrease as the intersection is farther away. Overall, the detector achieved high (> 0.9) precision and recall when the user was approximately 2–2.5 m away from an intersection. The second metric measures the distance between the user and the intersection when the first true positive detection occurs. Figure 3.4–B shows the results of the second metric. The letters in the x-axis are abbreviations of the intersection shape labels ("L" for Left, "R" for Right, "B" for Back, and "F" for Front). On average, the system was able to detect an intersection 2.47 m before reaching it.

Confirming the Existence of an Intersection

If a corridor has an uneven structure, such as an alcove, the YOLOv3 detector may detect it as an intersection, which is a false positive. As a result, the system may convey the wrong detection shape of the intersection to the user. Therefore, confirming whether the detected intersection is a true intersection or not is necessary. We implemented an algorithm to confirm whether the detected intersection is a true intersection when the user enters it. When the user is at a detected intersection, the system measures the furthest walkable cell beyond each side (left and/or right) of the intersection. If the distance between the nearest side of the bounding box and the walkable cell is beyond the threshold of ϵ m, the system confirms a path leading to that side. We set the threshold $\epsilon = 1.5$.

3.4.5 Interface of Corridor-Walker

Figure 3.5 illustrates the interface of Corridor-Walker. Based on previous studies, we designed our system to use TTS, spatialized audio, and vibration feedback. We kept

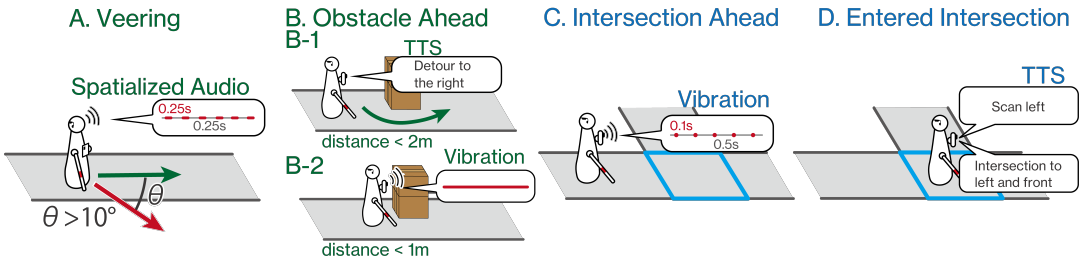


FIGURE 3.5: **Corridor-Walker Uses Multiple Modalities for Interface.** When the user is veering off the generated path, the system will correct the user's orientation with spatialized audio feedback. When an obstacle is detected within 2 m, the system will tell the user to make a detour. The system will also vibrate continuously when an obstacle is within 1 m. When an intersection is detected ahead of the user, the system will vibrate, then convey its shape using audio feedback when the user enters it.

the use of TTS minimal, as it may provide a high cognitive load to the user [164]. Thus replacing it with other suitable feedback may increase the efficacy of the system. TTS feedback is used to convey the shape of an intersection and tell the user to make a detour. Spatialized audio feedback is used to instruct the user to trace the generated path. Vibration feedback is used to notify the user of the existence of an intersection and to alert them to the imminent risk of collision. The system conveys auditory feedback (TTS and spatialized audio) through bone-conducting headphones and vibration feedback through the vibration of the smartphone.

Conveying Intersection Distance

The system vibrates when it detects an intersection ahead of the user. We used vibration to convey this information because the detector can detect an intersection 2.47 m ahead on average (Section 3.4.4), whereas feedback with TTS is too slow (the user would have reached the intersection during the TTS feedback). Previous studies have shown that users perceive more urgency at a lower interval [172, 173]. As we used vibration for both alerting the risk of collision and to notify the existence of intersections, different intervals were used for the two feedbacks. For alerting the existence of an intersection, we designed the vibration to be a single pulse vibration, whose pulse duration was 0.1 s and the interval was 0.5 s (Figure 3.5–Intersection Ahead).

Intersection Confirmation via Collaboration

Using the algorithm described in Section 3.4.4, the system adopts a collaborative interaction that uses the system's detection results as initial feedback and the user's input for confirmation. The purpose of this is to allow the user to confirm whether the detected intersection is a true intersection or not. When the user enters the intersection, the system tells the user to scan certain sides based on the detected shape of the intersection (*e.g.*, left and/or right) using TTS feedback (*e.g.*, the system will instruct the user to scan the left side if the label of the detected intersection is "Left, Back" or "Left, Front, Back"). Once the system determines that it is a true intersection (Section 3.4.4), the system will say which way the intersection leads to (Figure 3.5–Entered Intersection). Otherwise, the system remains silent. Note that users may additionally scan the side not indicated by the system if they suspect that the system's initial detection may be inaccurate. An example of the audio instructions when an intersection with the label of "Left, Right, Back" is detected is as follows:

TABLE 3.1: **Corridor-Walker User Study Demographics.** Participants' demographics are presented, including age, years of being blind, years of using smartphones, walking behaviour during the user study (classified as either far from wall or along wall), and their System Usability Scale (SUS) scores.

ID	Age	Gender	Total Blindness	Smartphone Usage	Walking Behaviour	SUS
P01	51	Male	40 years	7 years	Far from wall	82.5
P02	26	Male	11 years	6 years	Along wall	92.5
P03	52	Female	49 years	13 years	Far from wall	72.5
P04	61	Female	4 years	2.5 years	Along wall	80.0
P05	71	Male	5 years	6 years	Along wall	85.0
P06	34	Female	19 years	6 years	Along wall	75.0
P07	29	Male	19 years	10 years	Along wall	82.5
P08	35	Female	21 years	8 years	Along wall	87.5
P09	56	Male	5 years	10 years	Along wall	85.0
P10	63	Male	20 years	2.5 years	Along wall	75.0
P11	21	Male	21 years	8 years	Along wall	75.0
P12	53	Female	53 years	5 years	Far from wall	72.5
P13	34	Male	34 years	2.5 years	Along wall	92.5
P14	29	Male	24 years	10 years	Along wall	70.0

1) The user enters the intersection: *"Scan Left and Right"*; **2) User scans both sides, but there is a path only to the left:** *"(Intersection to) Left."*

Conveying Veering-Related Information

The system uses spatialized audio feedback to convey the correct orientation to the user (Figure 3.5–Veering). We used spatialized audio, as it has been shown that slight adjustments of orientation are challenging with TTS [9], but feasible with spatialized audio [157]. When the user veers off the path, the system provides feedback to rectify the user's orientation. If the user is facing the left/right while the user should be facing more to the right/left, the system will produce a sinusoidal tone (duration: 0.25 s, interval: 0.25 s, frequency: 400 Hz, Figure 3.5–Veering) from the right/left side of the bone-conducting headphone. When users can hear no sinusoidal tone from earphones, it means they are facing the correct orientation.

Conveying Obstacle-Related Information

When an obstacle is detected (Section 3.4.3) within 2 m of the user, the system will notify which way to make a detour through TTS feedback (Figure 3.5–Obstacle Ahead, Top Panel). For example, when there is an obstacle along the left side of the wall, the system will say *"Make a detour to the right."* If any obstacle, including the wall, is within 1 m in front of the user, the system will continuously vibrate (Figure 3.5–Obstacle Ahead, Bottom Panel). As the vibration with a shorter interval is capable of conveying an urgent situation [172, 173], we set the interval to zero, which means that the system will continuously vibrate until the user faces a safe direction.

3.5 User Study

To investigate whether users can employ collaborative interactions to identify intersections, and to examine whether the map-less system Corridor-Walker can support indoor corridor navigation, we conducted a user study in Waseda University's 121 Building, one representative target building for the system to address. We recruited blind participants to perform several tasks while using our system with a cane and compared the results to when the participants were using only a white cane, but not using the system. We use the term *system-aided* as the condition when the participants used both the system and a white cane to perform the tasks, and the term *cane-only* as the condition when the participants used only a white cane, but not the system. This user study was approved by the university's institutional review board (IRB).

3.5.1 Participants

Through an e-newsletter for blind people, we recruited 14 blind participants who travel independently on a daily basis. Table 3.1 shows the demographics of the participants. All participants mainly used white canes as their navigation aid and smartphones in their daily lives for more than two years, with a mean of 6.9 years and a standard deviation (SD) of 3.3 years.

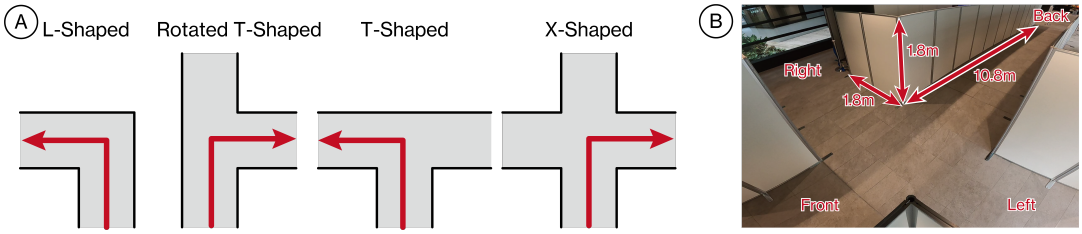


FIGURE 3.6: **Task Visualization of Turning and Identifying at a Single Intersection.** A) Intersections for task 1. B) The height and length of the corridors are shown.

3.5.2 Tasks and Conditions

Our user study involved the following three tasks.

Task 1: Turning and Identifying at a Single Intersection

This task was designed to evaluate intersection detection functionality and its collaborative interaction. In this task, participants were asked to turn in a specific direction (left or right) at an intersection, and then answer the shape of the intersection after each walk. We simulated intersections of different shapes using room dividers (Figure 3.6). For each walk, the participants were randomly placed between 6 m and 10 m before the intersection. Then we asked them to start the task from that location. The participants were notified before the task that they would be asked to answer the shape of the intersection after each walk.

Task 2: Obstacle Avoidance

This task was designed to evaluate path planning capability. In this task, participants were asked to walk through a 15 m straight corridor. We designed two routes:

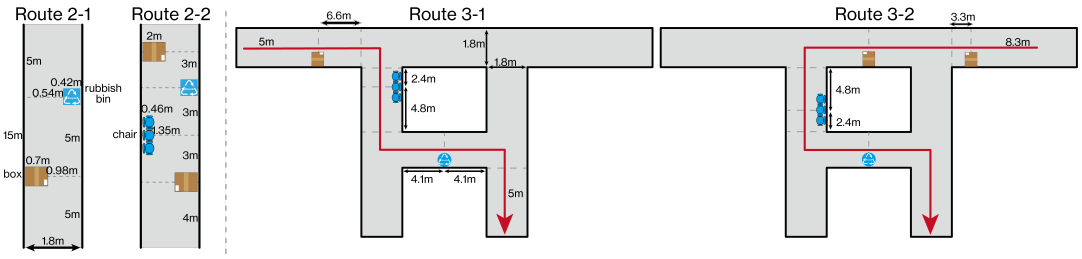


FIGURE 3.7: **Routes for Tasks 2 and 3**. Routes in Task 2 are designed to be topologically simple while containing multiple dense obstacles, whereas routes in Task 3 are designed to reflect more realistic settings with additional turns.

Route 2-1, which consisted of two obstacles, and **Route 2-2**, which consisted of four obstacles. We placed obstacles on the opposite side in turn (Figure 3.7). For example, a corridor may first contain an obstacle on the left side followed by an obstacle on the right side. We used a box, a chair, or a rubbish bin as obstacles, as shown in Figure 3.7. We randomly placed the participants 3 m or 6 m away from the route entrance, where the actual task started.

Task 3: Navigating Long Corridors with Obstacles

The final task aimed to evaluate Corridor-Walker in a practical setting. In this task, participants were asked to walk through a corridor with several intersections and obstacles. For this task, we used an existing corridor in our university. We designed two routes (Figure 3.7). **Route 3-1** had three intersections and three obstacles and was 37.4 m long. **Route 3-2** had four intersections and four obstacles and was 47.4 m long.

3.5.3 Procedure

We obtained informed consent from all participants, which was approved by the university's IRB. We first conducted a pre-interview, asking the participants about their daily experiences while navigating in indoor environments. Then, a training session with the system was conducted for 30 minutes. After the participants were accustomed to the system, they performed the three tasks in the main user study session. For each task, the participant walked all intersections or routes once in random order under system-aided and cane-only conditions. The order of the tasks under the two conditions was counterbalanced. The first half (P01–07) of the participants walked the intersections and routes listed in Figures 3.6–3.7 with the cane-only condition and the horizontally flipped intersections and routes of Figures 3.6–3.7 with the system-aided condition. The latter half of the participants (P08–14) walked the intersections and routes listed in Figures 3.6–3.7 with the system-aided condition and the horizontally flipped intersections and routes of Figures 3.6–3.7 with the cane-only condition.

After the main session, we conducted a post-interview. First, we asked the participants to rate a set of statements with the 7-point Likert items (ranging from 1: strongly disagree to 7: strongly agree). Each statement was asked for both the system-aided and cane-only conditions. These questions are illustrated in Figure 3.8, Q1–9. Then, we asked the participants to rate the system using the system usability scale (SUS) [174]. Finally, we asked open-ended questions to gather qualitative feedback on the system. During the experiments, we recorded videos of the participants

performing the tasks. These videos were used to calculate the metrics (Section 3.5.4). We also used the videos to classify each participant's walking behavior, whether they usually walked along the wall or walked far from the wall (*i.e.*, without relying on the wall) on the cane-only condition (Table 3.1). The whole study took 120-150 minutes in total for each participant. Each participant was compensated with $90 for their participation. To prevent the spread of COVID-19, the experimenter and participants covered their faces with masks and face shields.

3.5.4 Metrics

We used three metrics to evaluate our system. For each metric, the routes and the intersections that were flipped but had the same topological shape were named the same (*e.g.*, intersections whose shapes were "Left, Back" and "Right, Back" were both grouped as L-Shaped intersections).

Intersection Shapes Answered Correctly

For task 1, we measured the percentage of labels that the participants answered correctly. If the shape given by the participant after turning at an intersection matched the label of the actual shape, then the answer was considered correct. Otherwise, it was incorrect.

Task Completion Time

For each task, we measured the time to complete the task. For task 1, we measured the time it took to walk 5 m, from 4 m before (start) to 1 m after (end) each intersection. We started the timer when the participant reached the start and stopped the timer when the participant reached the end. For tasks 2 and 3, we measured the time it took to walk the route from start to end. We started the timer when the participant started walking and stopped the timer when the participant reached the end of the route.

Number of Contacts Made to Obstacles or Walls with a White Cane

For each task, we measured the number of times the participant made contact with obstacles or walls with their white cane by observing the videos taken during the experiment. For task 1, we only measured the number of contacts with the walls, as no obstacles were used.

3.6 Results

In this section, we describe the results of the experiments. First, we describe the daily experiences in navigating indoor corridors obtained through the pre-interview (Section 3.6.1), followed by the overall performance of Corridor-Walker (Section 3.6.2). Finally, we describe the qualitative feedback obtained from the post-interview (Section 3.6.3).

3.6.1 Daily Experiences of Participants in Navigating Indoor Corridors

To avoid obstacles, all participants agreed that they have to tap the obstacles with their white canes. Six participants mentioned that obstacles with hollow lower parts

(*e.g.*, chairs and desks) are challenging to avoid (P02, P05, P08, P09, P12, and P13), as the upper body may still collide. Meanwhile, P06, P11, and P12 commented that avoiding low-height obstacles (*e.g.*, boxes and rubbish bins) is also challenging because they cannot rely on their echolocation skills for detection.

As for locating an intersection, 12 participants (P01, P03–11, P13, and P14) mentioned that they walk along the wall and use a white cane to locate intersections, 10 participants (P01–04, P06–08, and P12–14) mentioned that they listen to the ambient sounds, and nine participants (P02–04, P06–08, and P12–14) mentioned that they perceive the flow of air. In a familiar place, in addition to the methods mentioned above, they also used a count of steps (P05) and intuition (P09, P13, and P14). Moreover, seven participants (P01, P03, P07, P08, P10, P13, and P14) reported that they had experienced walking past an intersection without noticing. Two participants reported that they had walked past an intersection when they were distracted (P01 and P03), and five participants (P07, P08, P10, P13, and P14) reported that they had walked past an intersection while avoiding obstacles. P08 described the relationship between intersections and obstacles as follows: **C3.1**:*"If obstacles or people are standing before an intersection, and because we have to avoid them, I lose track of my position and therefore may walk past the intersection"*[3] (P08).

Nine participants (P01, P04, P05, P07–10, P13, and P14) mentioned that it is difficult to walk straight in an indoor corridor. They mentioned that their main strategy is to listen to the echo of the sound from the nearby wall (P01, P03, P04, P06–09, and P12). P07 described the challenging experience of attempting to walk straight as follows: **C3.2**:*"It is difficult to walk straight. I think I am frequently veering or walking in a zig-zag shape"* (P07).

3.6.2 Overall Performance of Corridor-Walker

TABLE 3.2: **Participants Recognized Intersection Shapes More Accurately with Corridor-Walker.** This table reports the percentages of correctly identified intersection shapes in Task 1. Each intersection type is denoted concisely (e.g., "L" indicates an L-shaped intersection).

Intersection Shape	L	T	Rotated T	X
Cane-only	71.4%	21.4%	28.6%	0.0%
System-aided	92.9%	92.9%	100.0%	50.0%

Intersection Shapes Answered Correctly

Table 3.2 shows the percentages of intersection shapes answered correctly for L, T, Rotated T, and X-Shaped intersections for two conditions. Statistical analysis using the chi-square test at a significance level of 0.01 revealed that participants significantly answered the correct label on the system-aided condition in T ($p = 0.0004$), Rotated T($p = 0.0006$), and X-Shaped ($p = 0.009$) intersections. In the L-shape, the correct answers were high in both conditions and no significant difference could be observed ($p = 0.3$). The reasons why participants mislabeled the intersection with the system can be summarized: 1) Although the system did convey the correct

[3] All of the communications with participants were done in their native language. In this paper, we translated the communications to English and provide them in a quotation and italic, *e.g.*, *"translated content"*.

label of the intersection[4], the user answered another label (Occurred once with L-Shaped intersection, once with Rotated T-Shaped intersection and, three times with X-Shaped intersection), 2) the mapping of the system failed because the participant was holding the phone unsteadily, causing the YOLOv3 detector to output incorrect estimation results (Occurred once in X-Shaped intersection), and 3) the system correctly detected the X-Shaped intersection and instructed the participant to scan left and right, but the system did not tell the participant that it was an X-Shaped intersection because the participant only scanned in the direction of the intended turn (Occurred three times in X-Shaped intersection).

TABLE 3.3: **Number of Cane Contacts Made with Obstacles or Walls and Task Completion Time.** Mean, SD, and *p*-value of the Wilcoxon signed-rank test, comparing the system-aided and cane-only conditions. The symbols * and ** indicate the significance found at the levels of 0.05 and 0.01, respectively.

Task	Condition	Task Completion Time (seconds)			Number of Contacts			
		cane-only	system-aided	*p-value*	Contact with	cane-only	system-aided	*p-value*
1	L-Shaped	8.98±1.88	14.06±3.90	**0.0002****	wall	3.86±2.35	0.14±0.36	**0.004****
	T-Shaped	9.20±2.04	14.24±4.48	**0.0001****	wall	3.57±2.31	0.29±0.47	**0.006****
	Rotated T-Shaped	9.22±2.35	16.78±5.02	**0.0002****	wall	3.42±2.28	0.14±0.36	**0.002****
	X-Shaped	9.79±2.71	16.48±8.03	**0.0001****	wall	3.71±2.20	0.14±0.36	**0.002****
2	Route 2-1	18.55±3.61	23.46±7.42	**0.0006****	obstacle	1.28±0.73	0.50±0.52	**0.006****
					wall	3.14±3.61	0.57±0.94	**0.02***
	Route 2-2	20.91±4.85	28.50±7.67	**0.0006****	obstacle	2.21±1.42	1.35±1.00	0.08
					wall	1.86±3.18	0.62±1.01	0.2
3	Route 3-1	50.65±7.91	69.30±15.70	**0.0001****	obstacle	3.07±1.49	1.28±1.32	**0.01***
					wall	12.21±9.67	1.07±1.27	**0.003****
	Route 3-2	63.07±10.95	85.70±25.31	**0.0002****	obstacle	3.71±2.34	0.85±1.29	**0.002****
					wall	15.21±12.75	1.43±2.10	**0.002****

Task Completion Time

Table 3.3 reports the mean and SD of the task completion time. As this metric contains three factors that may affect the results, we first conducted a three-way analysis of variance (ANOVA) at a 1% significance level. Specifically, we compared cane-only and system-aided conditions, the order of conditions they started the tasks with, and whether the route was flipped or not. The analysis revealed that there was no interaction between all factors, and the cane-only and system-aided conditions were the only factors that affected the results. Therefore, to analyze the effect between the cane-only and system-aided conditions, we then separated the data based on the two conditions for each route and conducted the Shapiro-Wilk test at a 1% significance level. The test confirmed that normality could not be assumed for all metrics in each route. Also, as the three-way ANOVA revealed that flipping the route did not affect the result, the flipped routes can be assumed to be the same (*e.g.*, L-Shaped intersections that lead to the right and left can be assumed to be the same intersection). Therefore, we used the Wilcoxon signed-rank test to analyze the data. Our statistical analysis at a 1% significance level revealed that more time was required to complete all tasks using the system. This was because the participants tended to walk slower to follow the instructions and re-orient themselves while walking. Also, they took additional time to stop and scan the surrounding environment to confirm the shape of intersections when they were instructed to.

[4]This is verified by checking the system log.

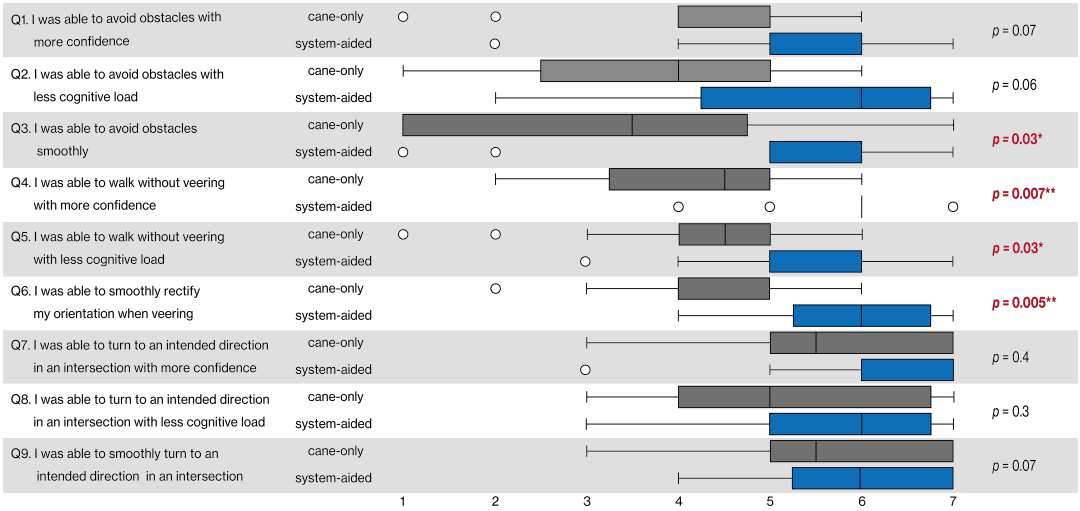


FIGURE 3.8: **Qualitative Evaluation with Seven-point Likert Questions.** The *p*-values calculated by the Wilcoxon signed-rank test are indicated on the left side of the figure. The symbols * and ** indicate the significance found at the levels of 0.05 and 0.01, respectively.

Number of Contacts Made to Obstacles or Walls with the White Cane

Table 3.3 shows the result of the metric. Based on the same reason stated in Section 3.6.2, we used the Wilcoxon signed-rank test to analyze the data. Our statistical analysis revealed that the system significantly reduced the number of contacts with walls and obstacles in all tasks except Route 2-2. Although the average value of the metric was lower with the system-aided condition in Route 2-2, the significance was not observed because each obstacle was placed only 3 m from each other (Figure 3.7), making the task challenging. Overall, the system enabled the participants to avoid obstacles while relying less on the wall to navigate the corridor.

Subjective Ratings

Table 3.1 shows the SUS scores for each participant. The mean SUS score was 80.5 (SD: 7.41). Figure 3.8 shows the results of the 7-point Likert items. Our statistical analysis using the Wilcoxon signed-rank test revealed that the system received significantly better ratings than the cane-only condition for Q3–6.

3.6.3 Qualitative Feedback

Appreciation to Corridor-Walker

Throughout the interview, we found that each participant found various aspects of the system advantageous. Twelve participants (P01–03, P05–08, and P10–14) felt positive about the obstacle avoidance function: **C3.3**:*"I was very impressed that I was able to avoid an obstacle without even knowing it was there. It is innovative that the system only signals when an obstacle is in front of me and stops notifying me once I start detouring around it"* (P06), and **C3.4**:*"Although I had to walk slower to listen to the feedback of the system, I was glad that I did not bump into an obstacle"* (P01). Nine participants (P01, P02, P04–06, and P08–11) especially felt positive about the correction of veering: **C3.5**:*"The system helped me to walk in a straight line. At first, I did not think it was*

necessary. However, it was useful because it helped me to walk in the middle when I cannot walk along the wall" (P09).

Thirteen participants (P01–11, P13, and P14) felt positive about the intersection detection function. Nine participants (P02, P04, P06–09, P11, P13, and P14) mentioned that they want to use this function to build mental maps. **C3.6**:*"Knowing that I am almost at an intersection means that I do not have to worry about running through it. By checking all the directions to which the intersection extends, I can discover that the road actually extends in another direction"* (P01) and **C3.7**:*"When I am walking with a white cane, I do not know which way the intersection actually leads to. With the system, I can perceive which way it leads to"* (P14).

Negative Feedback

Three participants (P03, P08, and P09) commented that the obstacle avoidance function of the system was insufficient because they naturally walk fast. **C3.8**:*"As I naturally walk quite fast, even if the system notifies me of an obstacle, my white cane hits the obstacle. I do not want to walk slower"* (P03). P12, born blind, found neither intersection detection function nor rectifying of orientation useful, as she could do both using only her echolocation skills. Two other participants (P03 and P13) also agreed that the correction of orientation was unnecessary. **C3.9**:*"I find intersection detection unnecessary because I can determine that I have entered an intersection only with my echolocation skills or by walking along the wall"* (P12) and **C3.10**:*"Since I think I can naturally walk in the middle, the correction of orientation is unnecessary. It is better if the sound comes from where an obstacle is"* (P12).

Smartphone Usage

All participants, except for P03, agreed that one strength of the system was that it requires only a single smartphone. **C3.11**:*"It is good that the system requires only one smartphone. I always have my smartphone when I go out"* (P08). However, 11 participants (P01, P03–05, P07–12, and P14) felt that holding the phone in their hands was a disadvantage. Especially four participants (P07, P08, P11, and P12) commented that maintaining the angle of the smartphone was challenging. **C3.12**:*"It was difficult to maintain the smartphone parallel to my orientation, as this system assumes that the orientation of the smartphone and the user is the same"* (P08).

3.7 Discussion

This section first discusses the two central research questions in this chapter, followed by an additional point that emerged through the study, and finally discusses the limitations of the system.

3.7.1 Did Corridor-Walker Allow Safer Navigation?

Overall, results suggest that although it took more time for participants to navigate in an indoor corridor (Section 3.6.2), Corridor-Walker successfully enabled all participants to navigate in an indoor corridor by assisting them to avoid obstacles and recognize intersections. The quantitative results (Table 3.3) suggest that the system enabled participants to make significantly less contact with obstacles and walls with a white cane. Also, the qualitative feedback (Figure 3.8) suggests that the system improved their experience while avoiding obstacles (Q3) and re-orienting themselves

(Q4–6). Comments from the participants suggest that they were glad to avoid obstacles without knowing that they were present (C3.3) and with less reliance on walls (C3.5).

3.7.2 Did Users Use Collaborative Interactions for Grasping Intersections?

Importantly, the recognized rate of intersection shapes (Table 3.2) and comments (C3.7) indicates that participants were able to identify intersections more accurately with the system. While the increased task completion time may seem negative, this supports the abovementioned important finding, as participants spent additional time on *scanning* - which we frame as collaborative interaction in this dissertation - so that the system can provide more accurate shape recognition. While there is room for improvement by making the scanning interaction faster, this presents a nuanced finding where simply increased task completion time is not always negative.

Furthermore, the intersection detection function of the system improved their intersection navigation experience by assisting them to prevent walking past an intersection unnoticed C3.6), and make a mental map (C3.7, Section 3.6.3). On the other hand, we did not observe statistical significance in questions about their experience when turning in an intersection (Q7–9). This is because Q7–9 mainly asked about locating and turning in an intersection that can already be performed with only a cane (C3.9) as well as the system.

3.7.3 Individual Preferences

Although Corridor-Walker enabled participants to safely navigate an indoor corridor, different preferences for functions and interfaces were observed. Some participants still made contact with obstacles when using the system (Table 3.3) although the sensing range of the system for obstacles was 2 m. P03 found the detection range of obstacles short, as she naturally walks fast (C3.8), whereas P01 did not find walking slower to be a disadvantage (C3.4). Thus, the default obstacle detection range does not need to be longer but should be adjustable for every user with a different walking speed.

P03 and P13, who had a high level of echolocation skills, did not find the intersection detection function (P13) or orientation correction function (P03 and P13) necessary (C3.8, C3.9, and C3.10). As they can naturally walk far from the wall (Table 3.1) by listening to the reflection of sound from the wall, they can locate an intersection when they enter it. However, P01, who also had high echolocation skills (Table 3.1), still felt positive about the intersection detection function as it can prevent the user from walking past it and tell the user the shape of an intersection, which is not supported by a white cane (C3.6). We observed that although blind people with high echolocation skills may walk without relying on walls and can locate intersections with only a white cane, they still have different individual preferences.

3.7.4 Limitations and Future Work

For the limitations of this study, the experiment was conducted in a limited environment with perpendicular intersections and a corridor with a fixed width. In actual usage, there may be an intersection that consists of five paths or gradual turns. As the current intersection detection function assumes that all intersections consist only

of perpendicular paths, the system may not detect such intersections. A more general labeling method for complicated intersections may allow us to create detection engines for a wider variety of intersections.

Also, the use of the system may be limited due to the sensing range of the LiDAR and its cost. As the sensing range of the LiDAR sensor is 5 m [165], the system can detect intersections only up to 3.0–3.5 m ahead (Figure 3.4). Moreover, the system assumes that both sides of the wall must be visible, thus limiting the use of the system in an open space such as a lobby or large foyer area. In such environments, both functions of the system will fail because the system cannot assign cost values to walkable cells for path planning and cannot extract features of the geometric structure of the environment to the grid map (*e.g.*, wall or corners in an intersection) for intersection detection. In addition, a smartphone with a LiDAR sensor is not yet common and affordable for all blind people. As smartphones are rapidly being improved in recent years, we believe that LiDAR-equipped smartphones with affordable prices and longer sensing ranges may appear and be widely adopted in the future, and these issues may be naturally solved along with the evolution of smartphones.

In terms of the ergonomics, 11 participants stated that there is a problem with how the smartphone should be held (Section 3.6.3). Since the optimal performance of the system requires users to hold a smartphone in an uncommon manner, they found it uncomfortable to hold it stably in front of them (C3.12). One failure in task 1 (Section 3.6.2, Reason (2)) occurred because of this reason (1.8%). Such a failure could become more pronounced because of fatigue of holding a smartphone if the person needs to use the system in the real world for longer periods of time. Despite this inconvenience, 13 participants still stated that the strength of this system is that it is implemented on a single smartphone (C3.11). As smartphone-based systems are highly accepted by blind people for their usefulness [154, 8], more longitudinal studies may provide insights into how to improve the ergonomic issue by further training and the extent of fatigue of holding a smartphone for a long period of time in real world situations. Therefore, collaboration with orientation and mobility (O&M) training communities could provide essential information and suggestions for designing methods to train the usage of such mobile navigation systems in addition to current methods such as white canes and echolocation for navigation.

As a next step, having investigated basic collaborative interaction and safe navigation, we aim to apply the system in an unfamiliar scenario. As explained in Section 3.3, Corridor-Walker is intended for use in situations where users already know the route. However, acquiring the route itself becomes an additional challenge for practical application in broader scenarios. In the following two chapters, we therefore consider two concrete scenarios. We set situations in which a blind user navigates to a destination either by analyzing a floor map or by using a route description obtained from surrounding people.

3.8 Conclusion

For the initial step of the map-less navigation systems, we present Corridor-Walker, a system that assists blind people in avoiding obstacles and recognizing intersections. The user study with 14 blind participants revealed that the system significantly reduced the number of contacts made with a white cane and enabled participants to avoid obstacles while relying less on the wall. By using collaborative interaction, which involves explicitly scanning the environment for the system, the system also

enabled participants to better recognize intersections compared to the case using only a white cane. For further design improvement, considering the different preferences raised by the participants is necessary. The next step will involve setting a concrete scenario and adding functions that allow the system to be used in unfamiliar locations.

Chapter 4

Snap&Nav: Smartphone-based Indoor Navigation System For Blind People via Floor Map Analysis and Intersection Detection

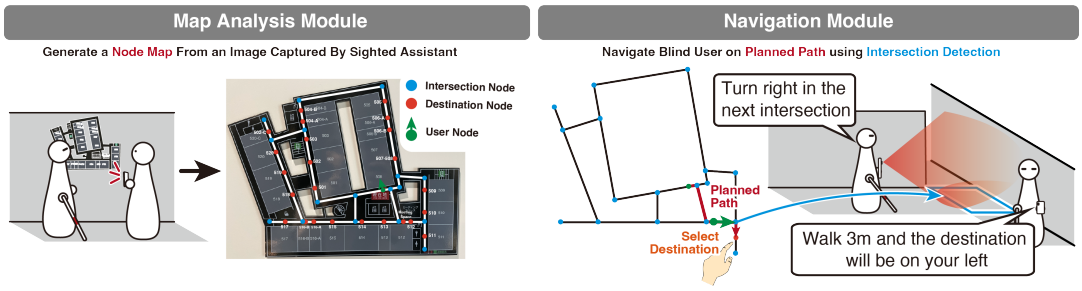


FIGURE 4.1: **Snap&Nav Provides Turn-By-Turn Guidance By Analyzing Floor Map Image.** The system first requires a sighted assistant to capture an image of a floor map that is commonly available at buildings. The system extracts a node map from the image by applying a map analysis algorithm. Then, the system plans a path to the selected destination by a blind user and navigates them to the destination by using an intersection detection algorithm.

4.1 Introduction

This chapter addresses the same indoor corridor-like environment with a smartphone, however, in an unfamiliar setting where blind people do not know the route to the destination. Navigation in an unfamiliar environment remains a significant problem for blind people. As they need a long-term familiarization of buildings for independent navigation with canes and guide dogs, they usually need to rely on sighted assistance by asking to accompany them to their destinations or asking them for the route [4, 107]. Their independence is further impeded by the fact that they need multiple assistance when navigating to another (multiple) destination.

The most common solutions include map-based navigation assistance systems [38, 9]; however, their usage is limited to places with prebuilt maps and supporting infrastructures. On the other hand, navigation systems that do not require prebuilt

digital maps [95, 175, 15] assist users by leveraging real-time sensing results to convey information about the environment, allowing users to decide their way at each decision-point. As these systems don't possess any pre-existing knowledge of the route users need to walk, users or systems must rely on external route information to be used. However, similar to Corridor-Walker [22], existing works either assumed that users already knew the route [22, 95] or relied on an experimenter to explain the route beforehand [15, 175], making them difficult to apply in unfamiliar settings.

One idea to source route information is to use floor maps, which are typically presented at the entrance of each building, as an information source for determining the destination. Therefore, in this chapter, by extending Corridor-Walker, we propose Snap&Nav, a navigation system that utilizes an image of a floor map in a building, which contains a walkable path along with names of possible destinations, as an information source of the environment (Figure 4.1). Firstly, to use floor maps for navigation, it is necessary to obtain the image of the floor maps for the system. As blind people may have difficulty in capturing an image of the target object by themselves [152, 111], we designed the system such that sighted people capture floor maps instead of blind users. Also, from the floor map image, it is necessary to extract information such as intersections, possible destinations, connection relationships, and the user's initial position and orientation for guidance to the destination. While intersections, destinations, and their connections can be extracted by utilizing image processing or computer vision algorithms on a floor map image [176], the users' position and orientation are not always apparent in all floor maps. Thus, we designed the system so that sighted assistants annotate the blind user's initial position and orientation. After a sighted assistant captures an image of a floor map, the system analyzes the image and creates a node map, which is a map involving the aforementioned information represented by nodes and connections. Secondly, by using the extracted node map, the system navigates blind people who hold the smartphone in their hands. To use the node map for navigation, it is necessary to continuously localize the user's position on the node map to provide turn-by-turn navigation instructions. Using the intersection detection algorithm of Corridor-Walker, the number and shapes of the detected intersections are compared with the intersections in the real world to keep track of which node or edge the user is currently on. Finally, to notify users that they have reached the destination, it is necessary to estimate the scale difference between the floor map and the real world. Therefore, by comparing the pixel distance between two intersections in the node map with their actual distance in the real world, the system estimates the scale.

We implemented Snap&Nav by prototyping two functionalities, namely map analysis and navigation, and conducted two user studies to investigate the following questions:

- Can sighted assistants capture and annotate the position and orientation of a blind user on the floor map, and are sighted people willing to perform this task for assist blind users?
- Can blind people reach their destination using Snap&Nav, and how does their experience compare to navigating with a cane?

The first user study was conducted with 20 sighted participants, where they captured an image of a floor map and annotated the position and orientation of a blind user. The study revealed that most participants were able to use the system without being accustomed to the system, but also revealed improvements such as the need for specification of how to verify whether the generated map is correct. Then,

the main study was conducted with 12 blind participants. We prepared two conditions: a *system-aided* condition where they navigated using the proposed system, and a *cane-only* condition where they navigated with a description of routes by sighted people. Participants were asked to navigate to multiple destinations in sequence. Throughout the study, we revealed that usage of our system enabled participants to navigate with increased confidence and reduced cognitive load, without affecting the task completion time. The participants generally appreciated the fact that they did not have to ask for route descriptions multiple times when using the system, which allowed them to gain more independence by relying less on sighted people. Additionally, ten of the blind participants expressed that they find it acceptable to involve sighted assistants, given the potential benefits they would receive.

4.2 Related Work

In this section, in addition to the related works reviewed in Chapter 2 — including navigation in unfamiliar buildings (Section 2.1.2), map-based navigation assistance systems (Section 2.2.1), and the works discussed in the previous chapter (Section 3.2) — we describe the prior floor map recognition method, which serves as an additional component compared to Corridor-Walker.

4.2.1 System Using Indoor Floor Map Analysis

Researchers have proposed various methods for creating navigation routes with edges and nodes from floorplans of buildings [177, 178, 179, 180, 181]. While floorplans accurately represent a floor's structure, these are not often available for public usage, which impedes assistance systems from using them for navigation purposes. Prior research proposed analysis systems for floor maps (*i.e.*, ones found on information boards in shopping malls or at entrances of buildings) to extract walkable areas [182] or localize the user's position in shopping malls [176]. Following previous research, we prototype a method to analyze floor maps, but for navigation purposes for blind people. To provide turn-by-turn navigation instructions, the system extracts information such as intersections, destinations, and their connections and generates a node map.

4.3 System Design

The proposed system, Snap&Nav, has a map analysis module and a navigation module (Figure 4.1). The map analysis module is aimed to be used by a sighted assistant, by having a blind user ask sighted people to use this module. Then, based on the analyzed map, blind users could select the destination and navigate using the navigation module. The system is designed to acquire the route to destinations from a floor map, which is commonly available in buildings. Thus, the advantage of this design is that it has the potential to be used in various buildings that have floor maps, without any preparation. To realize the design described above, we implemented the system on the iPhone 12 Pro, which is a smartphone equipped with a LiDAR sensor.

4.3.1 Map Analysis Module

To provide blind users with turn-by-turn navigation instructions, this module in the system creates a node map consisting of information of intersections, destinations, and the position and orientation of a blind user. Users of this module are sighted assistants (Figure 4.1). As they are expected to be asked to use the system on the first view, the interface of the system must be used without any prior tutorial. Thus, we design the map analysis module to provide voice instructions to sighted assistants. Firstly, the system instructs sighted assistants to capture the floor map image. Then, sighted assistants are asked to annotate blind users' position and orientation in the captured image by interacting with the system, as it is not always apparent in all floor maps. Then, the system processes the captured image to extract a node map consisting of the positions of intersections, destinations, and the blind user's position with their connections. Finally, the system asks sighted assistants to determine whether the connections in the node map match the information in the floor map. They can compare the node map displayed on the smartphone screen with the floor map. If they determine that the generated node map is not sufficient, they capture and annotate it again.

4.3.2 Navigation Module

Firstly, blind users can select the desired destination from the list of extracted destination nodes. Once the destination is selected, the system plans a path from the current user's node (Figure 4.1). To provide turn-by-turn instructions, the system tracks the user's position on the node map (*i.e.*, which nodes or connections users are in) by using the intersection detection algorithm to verify the shape of their current intersection and match it with the intersection nodes on the node map. To guide users to their destination, it's essential for the system to convey the distance they need to proceed from the final intersection. Therefore, the system calculates the scale difference between the node map and the real world by comparing the distance between two nodes in pixel space to the distance between two real-world intersections in meters. Once the scale difference has been obtained, it can offer instructions accompanied by the distance covered after that intersection. Note that this module is designed to navigate users globally (at a scale that requires multiple turns to reach the destination), not locally for obstacle avoidance. Therefore, obstacle avoidance is not part of the system's design requirements.

4.4 Implementation: Map Analysis Module

This section describes the implementation of the map analysis module, which is aimed to be used by a sighted assistant.

4.4.1 Interface for Sighted Assistants

Figure 4.2 shows the interface of Snap&Nav. When sighted assistants press the "capture floor map" button on the initial screen, the system activates the camera, and with voice feedback, the system instructs assistants to capture an image of a floor map so that it is placed in the center with minimum lighting (Figure 4.2-1). Then, assistants are instructed to annotate the position of blind users. Assistants can either tap or drag on the image to specify the position of blind users, at which point a green dot appears to indicate their location (Figure 4.2-1). When the annotation is

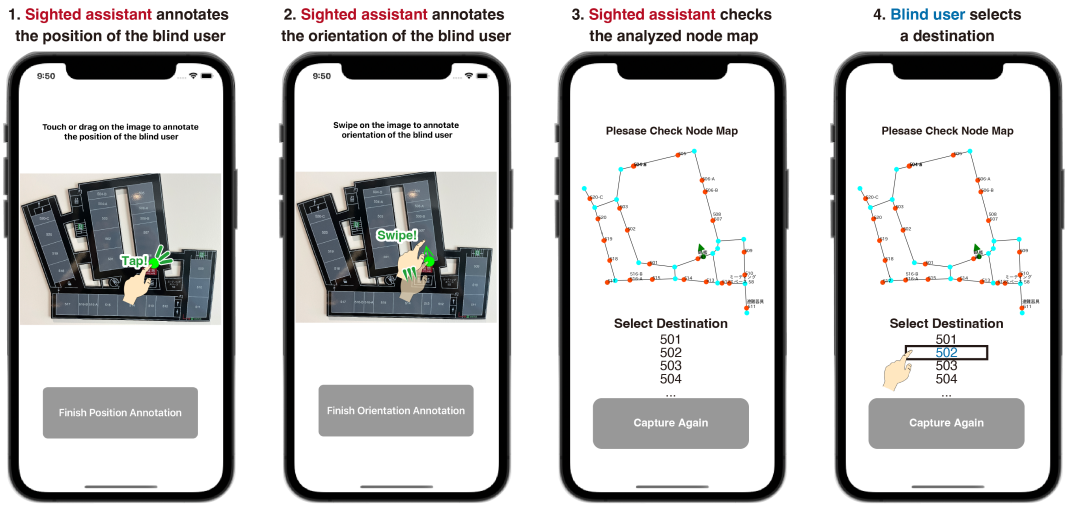


FIGURE 4.2: **Map Analysis Module Interface.** After capturing a floor-map image, the sighted user first annotates the blind user's location and then specifies the orientation with a swipe gesture. The system then runs the floor-map analysis algorithm and displays the result to the sighted user. If the output looks correct, the system is handed to the blind user for destination selection.

completed, assistants can tap a button placed on the bottom of the screen to complete the annotation process of the position. Then, the system provides voice feedback to instruct assistants in setting the orientation of the blind user at any angle by using a swipe gesture on the image. When assistants swipe, a green arrow pointing from the green dot will be displayed in the swiped direction (Figure 4.2-2). Finally, assistants can tap a button on the bottom to complete the whole annotation process.

The system sends the image to a remote server to apply a map analysis algorithm (Section 4.4.2). After the analysis, the system receives and displays the image of the analyzed node map from the server. Assistants can verify if the image is accurately analyzed (Figure 4.2-3). If not, assistants can repeat the process by pressing the capture image button.

4.4.2 Floor Map Analysis Algorithm

We prototyped a map analysis algorithm. Our algorithm creates a node map, which is a representation of a floor where each node corresponds to an intersection or a destination, and the connection between each node represents walkable pathways. The node map consists of three types of nodes: (1) a user node that represents the initial position and orientation of the blind user, (2) an intersection node, and (3) a destination node. Specifically, the node map will be obtained by applying the algorithm described below to an RGB image (resolution of 4032×3024) obtained by a smartphone camera on a remote processing server with RTX-3060 GPU with 8GB memory capacity.

The red circular and rectangular icon in Figure 4.3 represents the user's location. To prevent the icon from affecting the following image processing algorithm, we first mask out the icon, which indicates the user's position. To do so, we assume that the red color indicates the user's location in this study and mask out colors close to red in the image. Then, the image is binarized, and the connected component algorithm is applied to identify connected regions. The largest area, which is expected to represent the path or corridor, is extracted (Figure 4.3-1). Then, the skeletonization

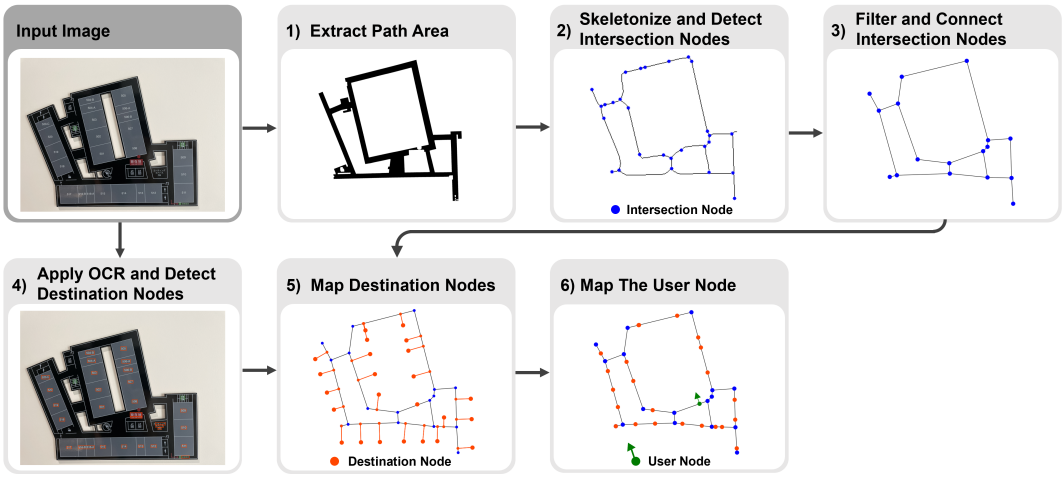


FIGURE 4.3: **Floor Map Analysis Algorithm.** Given an input image: (1) the path area is extracted using a connected-component algorithm; (2) skeletonization is applied to obtain the topological structure and nodes, which are assumed to represent intersections; (3) a filtering step removes nodes that do not correspond to valid intersections; (4) OCR is applied to the original image to extract potential destinations; (5) the midpoint of each OCR bounding box is mapped onto the previously obtained skeletonized map; and finally, (6) the system incorporates the annotated blind user's position and orientation data.

algorithm [183] is applied to the extracted path area. After that, Harris corner detection is applied to the skeletonized image to identify potential intersection nodes (Figure 4.3-2). The connections between these nodes are ascertained based on the connections presented in the skeletonized image. To eliminate extra intersection nodes that were accidentally detected, we filter out nodes with only two connections where the connection angle exceeds 140°, and get the intersection-only node map (Figure 4.3-3). Next, to find destination nodes in the floor map, optical character recognition (OCR) [184] is applied to the original RGB image. As a result, multiple bounding boxes with destinations' names are obtained (Figure 4.3-4). The center point of each bounding box is extracted to represent the location of each destination, which is then mapped to the nearest connection between nodes in the node map. At the same time, the system determines on which side of the path the destinations are located (Figure 4.3-5). Finally, we map the user node, whose location and direction were annotated by sighted assistants, to the closest connection (Figure 4.3-6).

4.5 User Study for Map Analysis Module with Sighted Participants

To investigate whether sighted assistants can capture and annotate a blind user's position and orientation on a floor map, and whether they are willing to perform this task, we conducted a user study evaluating the map analysis module with 20 sighted participants (16 male and four female). They were aged 22 to 31 years old (mean=23.8 and standard deviation(SD)=2.3). Eighteen participants were familiar with the experimental location, and two were unfamiliar. One aim of this study was to investigate if sighted assistants could use the system without being accustomed to it. This user study was approved by Waseda University's IRB, and informed consent was obtained from every participant before the study.

4.5.1 Tasks and Procedure

We took participants in front of five different floor maps, which are already available in the building. Then, we asked them to use the system, assuming they were asked to do so by a blind person. The experimenter, who acted as a blind person, stood within three meters of a floor map and faced towards it. Before handing the system, we gave concise instructions to capture the floor map image and annotate it by following the system's voice instructions. They were allowed to recapture and repeat the annotation until they felt confident with the generated node map. When participants finished the task, we asked them to return the system to the experimenter.

The floor map participants first capture using the system is the most important, as the map analysis module is designed for scenarios where sighted assistants are asked to use it by blind people. To ensure that each floor map is captured an equal number of times, we randomized the order of capturing each floor map, with each floor map being captured first by different participants precisely four times. Finally, we interviewed participants with questions on Figure 4.4. The whole study was recorded and took 30 minutes in total. Participants were compensated with 7$. Below, we refer to *first trial* as the first task of floor map capturing and *overall trial* as the all task of floor map capturing.

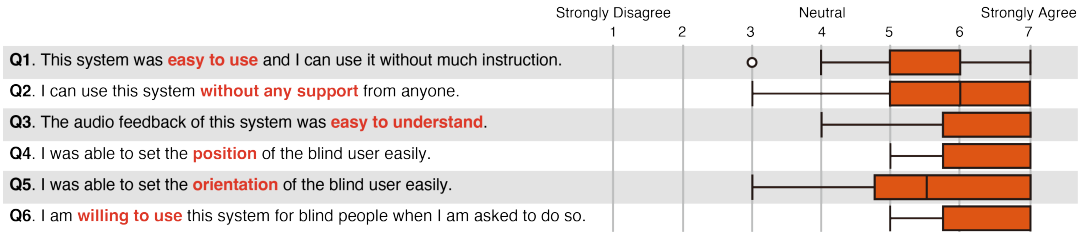


FIGURE 4.4: **Qualitative Evaluation with Seven Point Likert Items.** Questions and responses from our study with 20 sighted participants.

4.5.2 Metrics

Below, we describe the metrics adopted in this study.

Average Path Length Similarity (APLS)

To evaluate the performance of the map analysis algorithm, we used Average Path Length Similarity (APLS) [185], which is a standard metric for evaluating node maps such as ones generated from satellite images. The metric assesses the similarity between two node maps by comparing differences in their path lengths. This process involves identifying corresponding nodes between the predicted node map and the GT node map. The algorithm calculates the shortest path distances between all pairs of corresponding nodes using the Dijkstra algorithm [186], and records these path lengths. Subsequently, it computes the ratio of the length differences between these paths. If a corresponding node is absent in the predicted node map, resulting in a missing path, a maximum penalty of 1.0 is assigned. The final step calculates the sum of the differences between the predicted node map and the GT node map. This sum is then averaged across all nodes and subtracted from 1 to derive the APLS score. The APLS score ranges from 0 (indicates poor similarity) to 1 (indicates high similarity), and is defined as follows, $APLS = 1 - \frac{1}{N}\sum \min\left\{1, \frac{|L(a,b)-L(a',b')|}{L(a,b)}\right\}$ where

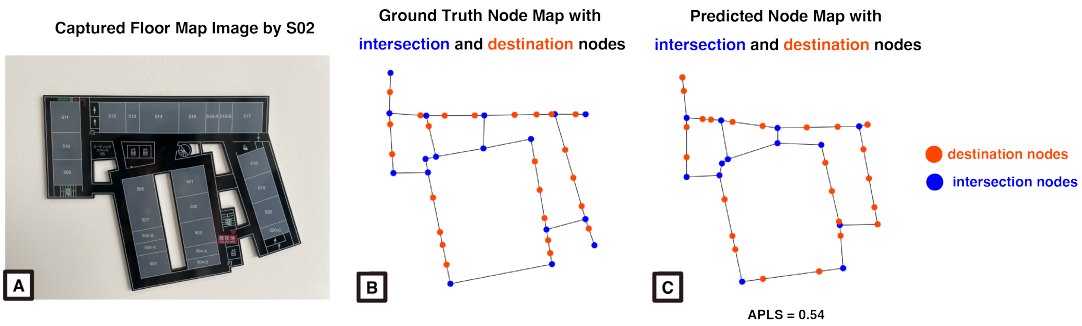


FIGURE 4.5: **Example of Captured Floor Map and Generated Node Map.** (A) a floor map image captured by S02, (B) its ground truth node map, and (C) generated node maps that contain both intersection and destination nodes.

$L(a, b)$ represents the path length between nodes a and b as computed by the Dijkstra algorithm in the node map. This metric is influenced by the nodes' topological connectivity and geographical positioning, as it is highly penalized when there is an absent connection and primarily measures the differences in path length.

Task Completion Time (TCT)

We measured the task completion time, which is the time to capture an image of a floor map and edit the user's position and orientation. We also define the *system process time (SPT)*, which is the time it takes to send the captured image to the server and generate the node map, and *confirmation time (CT)*, which is the time it takes for sighted participants to verify if the received node map is correct.

User Node Accuracy

We measured how accurately participants annotated the position and orientation of a blind user. We considered the position of a point to be correctly annotated if it was within 227 pixels of the ground truth (GT) location in the original image coordinate space. The value of 227 pixels was determined by imitating the minimum button radius of 22 points on the screen coordinate space for iOS devices [187]. As the screen width is 390 points for the iPhone 12 Pro and the image width is 4032 pixels, we calculated the value with the following equation, $4032 \times \frac{22}{390} \simeq 227$. We also defined the orientation as the correct orientation if the orientation is within 45 degrees from the GT orientation in the captured map. In a building where the system may be used, such as our experimental environment, floor maps may be installed on either side of the corridor wall. Thus, the orientation of a user standing in front of a floor map can be one of the two possible orientations. The system classifies the input user orientation into two categories based on which wall of the path the user node is orientated. If the error between the input and the annotation is within 90 degrees, the system determines the correct orientation. In this study, 45 degrees was used as a strict condition. We defined the GT position as the position of the floor map and the GT orientation as the side of the wall on which the floor map exists. The experimenter manually annotated the GT position and the orientation for the evaluation.

Subjective Ratings

We asked seven-point Likert-scale questions as shown in Figure 4.4. The questions were designed based on the system usability scale (SUS) [174] questionnaire. To fit within the time constraints of the study, we selected relevant questions from the SUS questionnaire, minimizing the total number of questions.

TABLE 4.1: **Map Analysis Algorithm and Participant Performance Evaluation.** Average Path Length Similarity APLS [185], task completion time (TCT), user node accuracy, and average number of recaptures per trial. For TCT, we report for overall time, system process time (SPT), and confirmation time (CT). We evaluated each metric for the first Trial and the overall trial.

	APLS [185] Intersection & Destination	Task Completion Time (Mean±SD [Seconds])			User Node Accuracy		Number of Recaptures per trial
		Overall	SPT	CT	Location	Direction	
First Trial	0.57	88.62±35.41	6.54±4.20	20.27±14.93	0.95	0.85	0.35±1.11
Overall Trials	0.56	62.92±28.40	6.42±3.06	14.91±12.68	0.99	0.95	0.21±0.60

4.5.3 Result

Below, we describe the result of the study.

Average Path Length Similarity (APLS)

Table 4.1 reports the average APLS of the generated node maps. The values did not differ between their first and overall trials, indicating how sighted participants captured images did not differ before and after getting accustomed to the system. In Figure 4.5, we provide an example of a captured image and its corresponding generated node map with its APLS. The average APLS value was 0.57 in the First Trial and 0.56 in the Overall Trial. The APLS values, which are close to 0.5, can be attributed to the misdetections of some nodes and deviations in node mappings, as illustrated in Figure 4.3-5. For instance, some nodes at the ends of corridors were missing because our algorithm mainly extracts intersection nodes with corner detection, and thus, corridor ends were not detected. Furthermore, while the algorithm successfully captured the overall structure of the node map, slight deviations in node placement from their actual positions were noted. These deviations also led to a decrease in APLS values.

Task Completion Time (TCT)

Table 4.1 shows the mean and SD of TCT. The mean value of the overall trial decreased compared to that of the overall TCT and CT of their first trial. Table 4.1 also shows the results of system process time (SPT) and confirmation time (CT) involved in overall TCT. The maximum CT was 63.94 seconds by S06.

User Node Accuracy

Table 4.1 shows the ratio of this metric. Generally, all participants were able to set the user's position correctly. On the other hand, three participants mistakenly set the orientation as they thought they were asked to set the orientation the blind person would be heading.

Number of Recaptures

Table 4.1 reports the mean and SD of the number of recaptures per trial. Seventeen participants finished the task without recapturing the floor map in their first trial. In overall trials, ten participants recaptured floor map images. Seven recaptures occurred in the first trials, and 21 occurred in the overall trials. Out of the ten participants, S18 showed the most confusion in their first trial. Firstly, S18 captured a floor map from a distance because S18 thought it was important to remove the light reflection. It caused the floor map in the image to be small. Thus, the system was not able to generate the appropriate map. As S18 thought that the cause of the failure was light reflections, S18 repeatedly captured the floor map in the same manner.

Subjective Ratings

Figure 4.4 shows the results of seven-point Likert-scale questions. For all questions (Q1–Q6), more than 17 participants gave positive scores (greater than 5). While all of them felt that they were able to set the position of the blind user (Q4), S03 and S19 felt that they were unable to set the orientation easily (they scored 3 points in Q5).

Qualitative Feedback

Aligned with the result in Q6, all participants stated that they are willing to use the system when asked by blind people, as it offers reliable assistance than explaining routes or guiding them: **C4.1**:*" It isn't easy to know if the route description is conveyed correctly. If the destination is far away, it takes time to convey the information, and I am also concerned about whether the information I provided is accurate. On the other hand, the system offers easy assistance just by capturing floor map images. And it doesn't take much time."* (S06) Also, S20 commented, assuming the scenario guiding a blind person. **C4.2**:*" I found it relatively easier than walking with them and guiding them to their destination. For example, when a blind person's destination is the exact opposite of the direction I want to go, or when the distance is very long, or when there is not much time available, I feel that this kind of application can reduce the burden on the person providing guidance."* (S20)

We also received negative feedback. Three out of twenty participants made errors in annotating the orientations of the blind user in their first trial. In this regard, six participants pointed out the ambiguity of the explanation for the orientation. **C4.3**:*" As for the orientation, I wasn't sure if I should input the direction the blind person was facing or the direction of the pathway."* (S14) Eight participants also commented about the difficulty of checking the node maps generated by the system. **C4.4**:*" In the case of a complex map, it would take a lot of time and effort to check if it is accurate. "* (S01) and, **C4.5**:*" The definition of whether the node map is good or not was not clearly stated. ... I thought it would be easier to check if the items to be checked were clearly indicated, for example, whether the room nodes are properly taken and whether the intersections are in place."* (S10)

4.6 Implementation: Navigation Module

This section describes the implementation of the navigation module, which is used by blind users after the map analysis module has been completed by sighted users.

4.6.1 Overview of Snap&Nav and Differences from Corridor-Walker

Snap&Nav supports destination selection, global route planning on the node map (Section 4.6.2), user position tracking on the node map using an intersection detection algorithm (Sections 4.6.3), and scale estimation during navigation (Section 4.6.4). Since Snap&Nav is built on top of our previous system, Corridor-Walker, we clarify its main differences here. The intersection detection method largely remains the same. However, the underlying model has been updated from YOLOv3 to YOLOv7. Obstacle avoidance functionality, which was a core feature of Corridor-Walker, has been removed in Snap&Nav because this study focuses on global navigation rather than local navigation (Section 4.3.2), and the effectiveness of local obstacle avoidance was previously investigated in Chapter 3. In contrast, destination selection, global route planning, user position tracking, scale estimation, and the updated user interface (Section 4.6.5) are newly introduced functionalities in Snap&Nav.

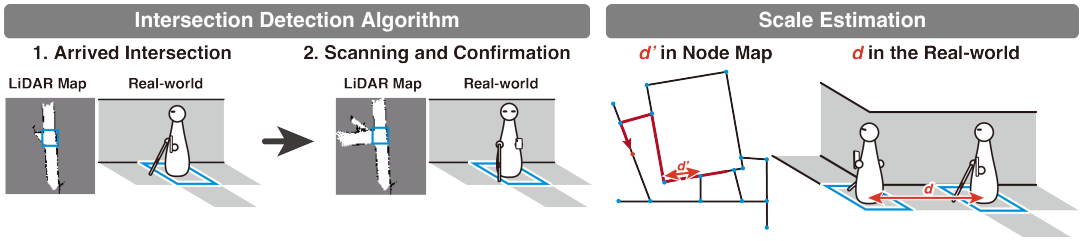


FIGURE 4.6: **Intersection Detection and Scale Estimation.** The system detects intersections and how the system estimates the scale difference of the node map with the real world. The displayed 2D occupancy grid map on the left panel is the actual grid map captured by the system.

4.6.2 Destination Selection and Path Planning

Firstly, the system lets a blind user select a destination on the node map from a list of destinations extracted in the floor map analysis (Figure 4.2-4) by using VoiceOver, the built-in screen reader on iOS. Then, the system employs Dykstra's algorithm [186] to plan a path from the current position, which is initially set to the user node, to the selected destination.

4.6.3 Tracking Users' Position Using Intersection Detection and Node Map

Intersection Detection and Confirmation

We use the intersection detection and intersection confirmation algorithm, which utilizes a method used in the Corridor-Walker [22]. The system generates a 2D occupancy grid map (*i.e.*, LiDAR Map) of the surrounding environment using the smartphone's LiDAR sensor and employs the YOLOv7 object detection model [188] to identify intersections, where the position of the bounding box indicates the location of the intersection and the label specifies its shape. The model is trained with the same dataset as presented in Section 3.4.4. Nine distinct intersection shapes, composed of combinations of the words Left, Right, Front, and Back, can be recognized (*e.g.*, Intersection on Figure 4.6–1 indicates "Left, Front, Back").

When an intersection has an uneven structure like an alcove, the system may mislabel it. Thus, the system checks the LiDAR map to confirm the shape of intersections by verifying whether each side bounding box contains a sufficient amount of walkable area. If the criteria for a specific direction are met, the system confirms

that the intersection leads in that direction. Users are instructed to scan specific directions in intersections to ensure the necessary features appear in the LiDAR map (Figure 4.6–2). In Corridor-Walker, the scanning direction was determined based on the detection result (*e.g.*, if the detected shape was "Left, Front, Back," the system instructed the user to scan only the left direction). In contrast, the proposed system instructs users to scan both sides of the intersection. This ensures that the detected shape is fully validated; for example, an intersection initially detected as L-shaped could actually be T-shaped if the unscanned side contains a corridor. Using the IMU sensor, the system monitors whether the user has scanned both sides by tilting the device by a predefined angle. Once the scan is complete, the system confirms the intersection shape and uses it to update the user's position.

Tracking Users Position

The system navigates the blind user to the destination by tracking their position on the node map. Every time the user reaches a detected intersection, the system compares the shape of the detected intersection with the shape of the next intersection in the planned path. The system calculates the direction of the paths, *i.e.*, the angle relative to the heading direction, in the node map and that of the detected intersection and compares them. If the error of these angles is within 40 degrees, the system determines that the intersections are matched, and the system updates their position on the node map and announces the next instruction.

4.6.4 Scale Estimation of Node Map

The system calculates the scale difference between the node map and the real world to convey users the distance they have to proceed once they have reached the first intersection. Every time the system passes an intersection, it calculates the distance d between the previous intersection in real-world scale and calculates the difference scale by $Scale = \frac{d}{d'}$, where d' denotes the distance between two intersections in the node map in pixel space coordinates. The system could calculate the distance to walk by multiplying the calculated scale by the length of sides in the node map.

4.6.5 Voice Feedback while Navigation

We designed our voice feedback that provides instructions on which direction to turn and the distance to proceed. While there are various types of voice feedback, including clock position instructions for tasks of lining up in a queue [8], simplified instruction to turn right or turn left for navigation[20, 114, 9, 41], and slight turn instructions (between 30 and 60 degrees) [189], we refer to the work by Kuribayashi *et al.* [22, 24], as they also convey intersection information in their task.

First, the system instructs users which way to proceed, along with the direction to proceed to the next intersection (*e.g.*, *"Face right, proceed forward, and turn left in the next intersection"*). Note that the initial direction for blind users to face (*e.g.*, *"Face right"*) is computed from the annotated orientation of a sighted assistant. When users have arrived at an intersection, the system instructs users to scan the intersection (*e.g.*, *"You have arrived at an intersection. Scan left and right for confirmation."*). If the scanned shape of the intersection is the same as the one users have to be at, the system instructs users to turn (*e.g.*, *"You are at an intersection to turn. Turn right"*). When the users have turned, the system provides users with the distance to proceed, along with the next direction turn. (*e.g.*, *"Proceed for 11 meters and face left.""*) Note

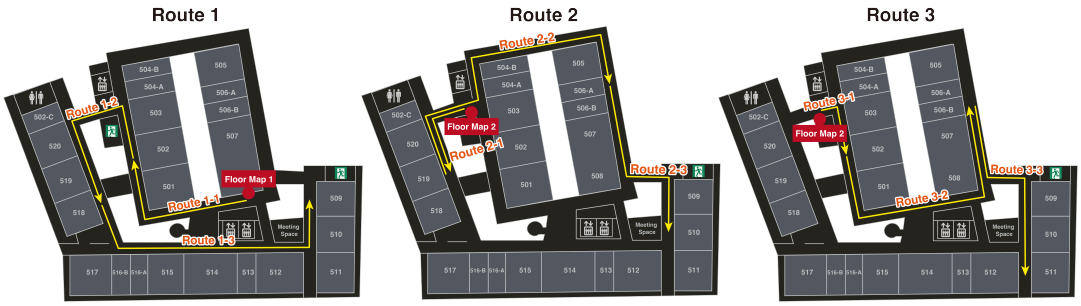


FIGURE 4.7: **Routes in Snap&Nav User Study.** To replicate the scenario of a blind person navigating to multiple destinations within one floor, each route contains multiple destinations.

at this point, as the system has already calculated the scale difference between the node map and real world, it could convey how much distance users should proceed. Finally, when users have arrived at the destination, the system notifies them of the way the destination is. (*e.g., "Face left. You have arrived at the destination."*) The system could provide instructions in real time as the whole process in the navigation module operates ten times per second.

4.7 User Study for Navigation Module with Blind Participants

To evaluate the navigation module of the Snap&Nav, we performed a user study in Waseda University's 121 building with 12 blind participants. Specifically, the user study was conducted to compare the system and when blind people use a white cane and walk based on the route described by sighted people. The scenario of this user study was as follows: *a blind user travels to an unfamiliar building and travels to multiple destinations.* Table 4.2 shows the demographic information of the participants. All participants are totally blind and use a white cane in their daily lives. While three participants had visited the experimental environment, they had no experience navigating the routes used in this user study. This user study was approved by the university's IRB, and informed consent was obtained from every participant.

4.7.1 Tasks and Conditions

The task of the study was to navigate several routes, each with three predefined destinations. Specifically, we prepared three routes, Route 1, Route 2, and Route 3, whose lengths were 122 m, 116 m, and 106 m, respectively (Figure 4.7). To mimic the scenario of moving to multiple destinations in an unfamiliar building, each route had three sub-routes, for example, Route 1 consists of Route 1-1, Route 1-2, and Route 1-3. For each route, they were asked to navigate sub-routes one by one and speak out to the experimenter when they reached the destination.

For comparison, we prepared two conditions, *system-aided* condition, and *cane-only* condition. Participants walked each route under two conditions, completing a total of six walks. In the system-aided condition, participants held their cane in their right hand and the system in their left hand. The experimenter, who acted as a sighted assistant, handed the smartphone equipped with the system in front of the floor map, assuming the situation of seeking an assistant to the sighted person and capturing the floor map has already been completed. Then, participants walked

three sub-routes within a single route independently using the system. This assumption was explained to the participants prior to the task. To focus on the evaluation navigation aspects, the system used two node maps with a high value of APLS obtained in the user study with sighted participants. The APLS value of the node map used in Route 1 was 0.63, and the value of the node map used in Route 2 and 3 was 0.59. In the cane-only condition, participants only had their cane. The experimenter provided a description of each sub-route at each starting point of sub-route. (*i.e.*, descriptions were given three times per route) When participants believed they had arrived at their destination, they verbally indicated their arrival to the experimenter. The descriptions of the route consisted of an accurate number of turns and distances they had to walk. They were allowed to ask the experimenter for the route. In such a case, the experimenter would explain the route from their current position to destinations.

4.7.2 Procedure

We first explained the purpose of the study and conducted 20 minutes interview asking about their demographic information and daily experience when navigating unfamiliar buildings. We then introduced the proposed system and participants practiced using the system in a test area for 30 minutes to get familiar with the system. For the training session, participants navigated through five pre-determined routes using the system. Some participants navigated an additional route if they needed to familiarize themselves more with the system. Then, participants were asked to conduct the main task. In order to counterbalance potential order effects, participants systematically rotated through the experimental conditions. For the first half of the participants, the progression began with Route 1 with the system-aided condition, and subsequently alternated conditions with each successive route (*e.g.*, B01 walked Route 1 with system-aided, then Route 2 with cane-only, Route 3 with system-aided, Route 1 with cane-only, Route 2 with system-aided, Route 3 with cane-only). The second participant in this group started with Route 2 with the cane-only condition, maintaining the alternating route and condition order. The translation of the route and condition was maintained until the sixth participant. For the latter half, this order was reversed (*e.g.*, Route 3 with cane-only, Route 2 with system-aided, Route 1 with cane-only, and so on). Finally, after the main task, we conducted a post-interview, asking them questions regarding the usability of the system with both open-ended questions and questions using Likert scores [174]. The whole study was recorded, and the study took approximately 135 minutes. Participants were compensated with 25$ per hour.

4.7.3 Metrics

Below, we describe the metric used in this study.

Task Completion Time (TCT)

We measured task completion time, which is the time to complete routes. Task completion time was recorded both for the time they travel the whole route and the time they travel from one destination to another. A timer was started when participants started navigating and was stopped when they stated their arrival at the destination. We also measured the time participants scanned at each intersection, by observing the recorded video.

TABLE 4.2: **Snap&Nav Participant Demographics.** The table reports each participant's age, gender, total years of blindness, years of smartphone usage, and SUS score.

ID	Age	Gender	Total Blindness	Smartphone Usage	SUS
B01	58	Male	18 years	8 years	92.5
B02	37	FeMale	23 years	9 years	95
B03	31	Male	21 years	10 years	90
B04	50	Female	13 years	13 years	92.5
B05	28	Male	14 years	9 years	95
B06	55	Female	51 years	12 years	92.5
B07	48	Male	8 years	7 years	77.5
B08	50	Female	5 years	5 years	87.5
B09	38	Female	37 years	6 years	75
B10	48	Female	45 years	10 years	100
B11	50	Female	15 years	10 years	70
B12	57	Male	3 years	20 years	72.5

Distance to Destination Area

Our system is designed to offer global turn-by-turn navigation instructions to reach a destination, which in this study's context, is a specific area. It is not our goal to provide last-few-meters guidance, such as guiding users to the exact entrance of the room [114, 110]. Thus, to evaluate the accuracy of our navigation system, we measured the distance between the point where participants stated their arrival and the destination area where the rooms are located. If participants stopped within the width of the rooms, we considered this metric as zero. However, if they walked past the destination area, we measured the distance from the end of the room to where they stopped.

Subjective Rating

We conducted subjective ratings to quantitatively assess the usability of the system (Figure 4.9). As illustrated in Figure 4.9, we evaluated confidence and cognitive load for each functionality by comparing system-aided and cane-only conditions. Additionally, we assessed ease of understanding, usefulness, and appropriateness for the system-aided conditions. To design the questionnaire, we referred to the question presented in the previous research [52, 47, 141, 24, 11]. We note that while some questions for the cognitive load can be measured using the NASA-TLX questionnaire [190], we adopted the aforementioned design method to minimize the total number of questions and fit within the time constraints of the study.

4.8 Results

Below, we present the detailed results for the user study with blind participants

4.8.1 Overall Performance

Task Completion Time

Figure 4.8–A shows the mean and 95% confidence interval (CI) for task completion time for each sub-route. We conducted the Shapiro-Wilk test for the normality of

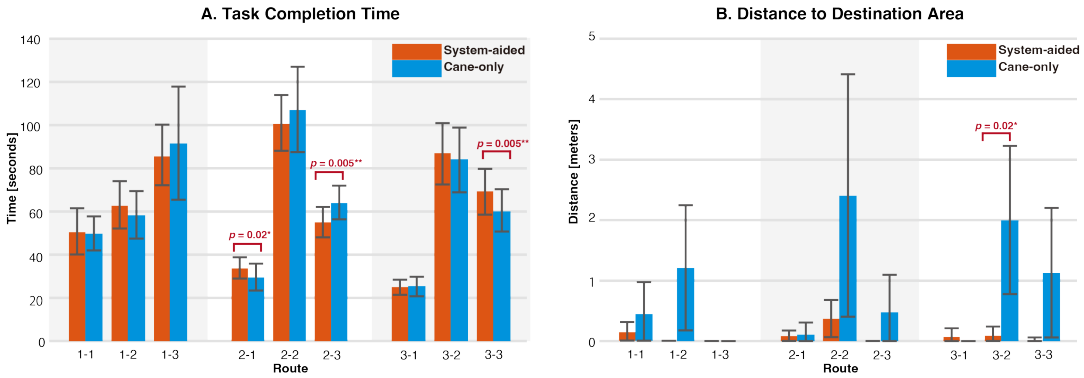


FIGURE 4.8: **Sna&Nav Could Guide Users Closer to The Destination Area While Preserving Task Completion Time.** Bar graph shows the distribution of task completion time and distance to the destination area for each sub-routes. The error bar shows the standard deviation.

TCT for nine sub-routes. Out of nine sub-routes, normality was not confirmed for three sub-routes. Thus, we used the nonparametric test, the Wilcoxon signed-rank test, to compare the metric between system-aided and cane-only conditions. We compared TCT for two conditions using the Wilcoxon signed-rank test and revealed that the system-aided condition significantly took a longer time for Route 2-1 and Route 3-3 and a shorter time for Route 2-3 ($p < .05$ for all Route 2-1, 2-3, and 3-3), compared to the cane-only condition. In the cane-only condition, some participants had difficulty finding intersections and destinations and sometimes got lost. B04 got lost in Route 1-3, and B09 and B10 got lost in Route 2-2, resulting in the cane-only condition having larger confidence intervals for task completion time for these routes. Also, Table 4.3 reports the average scanning time in each sub-route.

Distance to Destination Area

Figure 4.8–B shows the result of this metric. We conducted the Shapiro-Wilk test on this metric for nine sub-routes. Out of nine sub-routes, normality was not confirmed for eight sub-routes. Thus, we used the nonparametric test, the Wilcoxon signed-rank test, to compare the metric between system-aided and cane-only conditions. There were no significant differences for all routes except Route 3-2 ($p < .05$). This is because some participants had no difference in this value, zero, regardless of the condition. Still, the Figure shows that the system-aided condition produced generally smaller mean values and confidence intervals than the cane-only condition. This can be attributed to the system's ability to guide participants within the destination area and prevent them from making significant navigation errors. In the cane-only condition, while we provided accurate distances in the route description, seven participants made more than three meters of navigation errors. For example, B04 arrived 9.6 meters, and B08 arrived 7.6 meters away from the destination.

Number of Times Asked for Route Description and Subjective Rating

Table 4.3 shows the average number of times participants asked for route descriptions. Participants did not ask for the route descriptions in the system-aided condition.

Figure 4.9 shows the results of subjective ratings. We performed the statistical analysis using the Wilcoxon signed-rank test for Q1–Q6 and observed significance

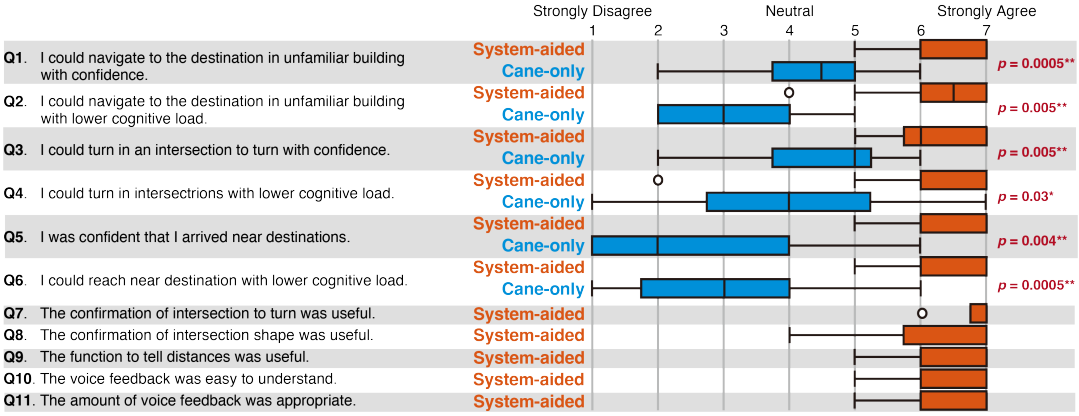


FIGURE 4.9: **Snap&Nav Could Guide Users with More Confidence and Lower Cognitive Load.** Questions and seven-point Likert scale responses from our study with blind participants. Responses marked with * indicate $p < .05$ and ** indicate $p < .01$ significant difference between the systems when applying the Wilcoxon signed-rank test .

TABLE 4.3: **Average Route Requests and Intersection Scanning Time.** The table reports the average number of times participants requested route descriptions, and the average scanning time per intersection.

		Number of Intersecitons	Length of Route [Meters]	Average Times Asked per Route	Average Scan Time Per Intersection [Seconds]
Route 1	1-1	1	32	0.75	4.56
	1-2	2	35	0.67	5.38
	1-3	2	55	1.00	6.14
Route 2	2-1	1	16	0.67	6.84
	2-2	4	64	2.33	4.47
	2-3	2	36	1.00	3.68
Route 3	3-1	1	12	0.25	5.50
	3-2	2	54	1.00	4.90
	3-3	2	40	0.67	4.17

for all questions, indicating that the system-aided condition was rated higher than the cane-only condition ($p < 0.05$ for all Q1–Q6). Moreover, all participants gave positive scores (greater than 5) to our system for Q7, Q9–Q11. For the ability to tell the shape of intersections (Q8), most participants except B03 and B11 gave positive scores. Also, Table 4.2 shows the SUS scores.

4.8.2 Qualitative Feedback

Compared to their usual strategy of navigating unfamiliar buildings, ten participants appreciated the system's design, which involves the capturing process of a floor map image by sighted assistants, as it may enable them to navigate to multiple destinations independently: **C4.6**:*" (Route description by sighted people is) fine if all you have to do is go to the room. However, you may need to leave the room and move around the building. In such cases, if I have a picture of a floor map taken by a sighted assistant, I may be able to move around independently. I think it is a good idea because we can reduce various costs just by having the photos taken."* (B10) and, **C4.7**:*" Asking where I want to go in the first place is, of course, hard and stressful, but in the end, I catch people again and*

ask, 'Where is the entrance?' is also stressful. This system does not require that, so it is good that I don't have to ask for help all the time." (B03) In addition, regarding their cognitive load, participants appreciated that they did not have to remember the route to their destination: **C4.8**:*" For example, I can remember an explanation of just going straight and turning right, but I can't remember if there is some further explanation. The system was very good because I didn't have to remember, and I could leave it to the system to guide me."* (B08)

On the other hand, B06 and B09 disagreed with the design that required them to ask the sighted people to capture the floor map. B06 and B09, as well as B07, expressed concern about handing their smartphones to others: **C4.9**:*" My iPhone is really precious to me, so I honestly don't like asking someone I don't know to take a picture with it."* (B06) B09 and four other participants (B01, B07, B08, and B11) also expressed concern about relying on sighted people who did not know when using the system: **C4.10**:*" I think getting assistance from sighted strangers easily is a challenge in this system. (When I talk to strangers) I don't know who we are talking to or what kind of people we are talking to."* (B09)

Many participants appreciated the ability to tell the distance. **C4.11**:*" The system told me when I arrived, "Please face the right" or "Please face left." So, I felt the distance was right. With a white cane, thinking about how many meters I have gone, I walk 1m, 2m, 3m, 4m...and so on. I can only walk with the feeling that this would be about 8 meters. Thinking about meters is extremely tiring because I'm using my nerves to walk."* (B04) Related to Q3, Q4, Q7, and Q8, ten participants provided a score above 4, commenting positively on the function of notifying them of the information on intersections. **C4.12**:*" Usually, I am unsure where intersections are, so I have to follow the wall to find the edge of the wall. Intersections are scary for those who cannot see. So I was impressed that the system told me the exact location of the intersection."* (B08) In particular, ten participants provided positive feedback towards systems feedback, conveying the shape and directions of intersections. An example of the feedback to this feedback was: **C4.13**:*" Using a cane, I cannot determine if this is truly an intersection or just a hollow part of the wall. If the system knows that we are at an intersection, it tells me to "go to left", and then I can understand that I need to turn left without verifying with my cane."* (B12)

4.9 Discussion

4.9.1 Acceptance of Snap&Nav

Throughout the study with sighted participants, most of them were able to use the system without being accustomed to the system, and appreciated that the system eliminates the need to explain routes (C4.1) and the need to guide blind people to their destination (C4.2). Crucially, all participants answered they were willing to use the system when asked by blind people (Q6 in Figure 4.4), implying that our collaborative design involving sighted assistants may be appreciated.

Throughout the study with blind people, participants appreciated that they were able to navigate independently in the unfamiliar building compared to their daily experiences. Usually, blind people have to ask others multiple times to navigate multiple destinations (C4.7) and memorize the description of the route by sighted people (C4.8). In contrast, the design of our system allows them to independently navigate to multiple destinations within a floor once a floor map image has been obtained, without memorizing the description of the route. Therefore, with the proposed system, participants were able to complete all tasks without asking for a route (Section 4.8.1). Furthermore, although the system required users to scan at every

intersection, adding roughly five seconds to the process, they were satisfied with the overall experience the system provided (Figure 4.9). Most importantly, ten participants expressed that the total benefits of the system outweigh the inconvenience of asking for assistance with image capturing once before the navigation (C4.6), indicating the potential acceptance of the design the system, which involves sighted people in the intermediate step, by blind people.

While recent research has primarily aimed at reducing user effort in navigation systems through automation [9, 46, 47, 48, 191], our approach emphasizes interaction with sighted assistants and the system. We achieve map and sensor infrastructure (*e.g.*, BLE beacons) with fewer solutions by engaging sighted people in acquiring floor map images and blind users in scanning environments at intersections. Although this approach might initially cause reluctance among both sighted assistants and blind users, our experimental results demonstrate its potential acceptance by both groups. We note that future solutions may adopt an approach from Teng *et al.* [192], which would seamlessly connect blind users to nearby sighted assistance. Overall, by employing a design that involves assistance from sighted people and scanning interaction from blind users, we achieve the first step towards scalable and map-less navigation for blind people in potentially diverse buildings.

4.9.2 User Experience of Map Analysis Module

Sighted participants were able to use the map analysis module without being accustomed to it. Following the system's voice instructions, participants were able to capture floor maps that appeared wide in the image with minimum lighting, resulting in a node map with minimum errors, as illustrated in Figure 4.5. On their first trials, they were able to complete the task with an average time of 88.62 seconds and felt they were able to use the system easily (Q1–2 in Figure 4.4). Out of 100 overall trials, most participants successfully annotated position and orientation, with only one position error and five orientation errors. The sighted participants agreed that they were willing to use the system when asked by blind people (Q6 in Figure 4.4). Meanwhile, we observed concerns from participants. One major concern was the difficulty in determining whether node maps can be used for navigation (C4.4 and C4.5). After generating a node map, the system instructs the user to check the node map, but it did not instruct what specifically should be checked. Therefore, the future system should display the criteria to be checked, such as the correctness of the locations of nodes and their connections.

4.9.3 User Experience of Navigation Module

Using the system, blind participants were able to arrive closer to their destinations with increased confidence and reduced cognitive load, without affecting the task completion time. While in the system-aided condition, it took time to scan at intersections (Table 4.3), in the cane-only condition, it also took time for participants to complete the task for reasons such as the difficulty in finding intersections and destinations (Section 4.8.1). As a result, we did not observe significant differences for all routes between the TCT of the system-aided condition and the cane-only condition (Figure 4.8). This indicates that the participants mostly maintained their usual walking speed while using the system for navigation. The distance to the destination area of the system-aided condition was less than one meter on average, and its confidence intervals were smaller compared to the cane-only condition (Figure 4.8). It indicated the system navigated participants to the destination with small errors.

Furthermore, the system enabled participants to navigate with more confidence and lower cognitive load (Q1–6 of Figure 4.9), which is due to two functionalities: announcement of the existence of an intersection, and announcement of distances. All participants provided feedback that the system's announcement of distances via scale estimation (Section 4.6.4) and announcement of the entrance of an intersection via intersection detection (Section 4.6.3) helped reduce their cognitive load, as they no longer had to keep track of how far they had walked (C4.11) and struggle finding intersections (C4.12). While we designed simple voice feedback to convey the shape of intersections or directions to turn, ten participants appreciated it (C4.13). Considering the above, we conclude that the system enhanced their navigation experience to the destination.

4.9.4 Concern of Dependence on Sighted Assistants

While the system design was appreciated by blind users, several considerations need to be made for the design of the system. Two participants disagreed with the design of the system, which incorporates sighted assistants' help in the system usage flow. They were concerned that they had to ask sighted people to capture a floor map image. One of them mentioned that they would not like to hand their smartphone to others as it is expensive and precious to them (C4.9), and the other expressed concern about relying on strangers (C4.10). To address the first concern, we may adopt a design to have sighted assistants capture floor map image with their own smartphone and send it to blind users' devices (*e.g.*, via Airdrop[193]). To address the second concern, an alternative strategy could involve blind users scanning the environment while the system attempts to localize itself from the information acquired through it. The system may analyze features such as the shapes of intersections or the names of stores on signage [116]. Such information could serve as landmarks, allowing us to refine and apply localization techniques previously established in research [194].

4.9.5 Limitation and Future Work

This user study design and the system had several limitations. Firstly, the experimental location contained only simple 90-degree turns. However, in real world environments, there may be complex-shaped intersections, such as those with large open spaces or Y-shapes, which the current system is unable to detect. Although more advanced intersection detection algorithms have been proposed [83], these approaches typically require more sophisticated sensors, such as 360-degree LiDAR, and are therefore difficult to apply directly to smartphones at present. Also, the optimal feedback method for complex-shaped intersections may differ from the straightforward instructions we used in this study (*e.g.*, "left" and "right") Therefore, further advancements in smartphone-based intersection detection, as well as effective methods for conveying complex intersection shapes to users, are required.

Secondly, the floor map analysis module also does not handle users' orientation in large open spaces. While detailed orientations are required in such spaces, the system only classifies the input user orientation into two orientations, which in this study used the threshold of 45 degrees to evaluate. The system, however, can still incorporate orientations annotated by sighted assistants, as it allows sighted assistants to annotate precisely. Moreover, the system could utilize the localization method described in Section 4.9.4 for localizing and ensuring precise orientation.

The performance of floor map analysis is the core element of this system. We prototyped the floor map analysis algorithm and evaluated it with five floor maps

in the experimental location, as the main focus of our study was on validating the system design, not the accuracy of general floor maps. The floor map analysis algorithm involved certain assumptions. First, the red region was assumed to represent the current location. Second, the largest area identified by the connected component algorithm was assumed to represent the path area of the floor map. However, when considering the practical usage in the real world, the floor map analysis algorithm needs to be more generalized and evaluated on other floor maps. For future work, we aim to develop a more generalized algorithm. In addition, the system assumed that the scale of the captured floor map was the same over the entire image. The scale of floor maps may not always be accurately presented. Thus, the algorithm also needs to take into account scale differences for real world deployment.

Finally, there were limitations in the participant recruitment. In the first user study with sighted participants, the average age of the participants was 23.8, and the maximum age was 31. In addition, 18 of the 20 sighted assistants were familiar with the building. Therefore, it is unclear how other generations and people unfamiliar with the building would evaluate the system's usability. Thus, we aim to conduct the study with a broader age range in the future. In the second user study with blind participants, three participants had previously visited the experimental location for the previous study. Thus, the three participants may have had a bias, such as positive impressions of the study.

4.10 Conclusion

Through the development and user study of Snap&Nav, this research demonstrates the potential acceptability of involving sighted individuals in the system workflow, as both blind and sighted participants showed positive responses. Specifically, this design targets scenarios in which blind users navigate unfamiliar buildings using a map-less navigation system. The results also highlight the need for further development and rigorous evaluation of the floor map analysis algorithm. One major limitation of the current system lies in the intersection detection algorithm, which is inherited from Corridor-Walker. As a result, the system can detect only simple intersections in corridor-like environments. This limitation is partly due to the reliance on a smartphone platform, which offers limited sensing capabilities and computational resources.

Chapter 5

PathFinder: Designing a Map-less Navigation System for Blind People in Unfamiliar Buildings

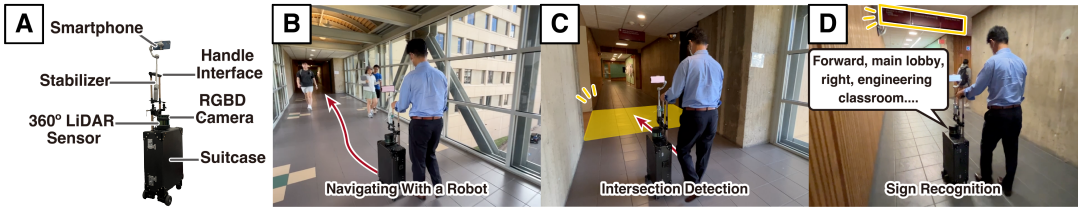


FIGURE 5.1: **PathFinder Overview.** We present PathFinder, a map-less navigation system that can navigate blind people in unfamiliar buildings by detecting intersections and recognizing signs.

5.1 Introduction

To explore further the possibility of map-less systems, this chapter extends the platform from a smartphone to a navigation robot [11, 60, 195] that can autonomously guide blind users using motorized wheels and provide rich environmental information through multiple sensors, including stable RGB cameras and a 360-degree LiDAR sensor. Previous studies [11, 60] have shown that the use of autonomous navigation robots is effective, as blind people only need to follow it, and therefore it can increase users' confidence and decrease their cognitive load when navigating buildings. This creates an advantage that users can concentrate on the decision-making process and the surrounding perception required for collaborative interaction. Building on this advantage, we aim to develop a map-less navigation system that imitates the collaborative interaction between guide dogs and blind people, where the robot (guide dog) takes over mobility, and the user decides where to go based on surrounding perception. Still, as guide dog interactions are used when users do not know the route, our system needs to convey various information and have the user decide their way, and blind people's decision-making capability remains underexplored.

This chapter focuses on scenarios in which blind users navigate public buildings, such as universities or hospitals. While floor map analysis approaches presented in Snap&Nav could be applied, resulting in autonomous navigation for robots, in many cases, floor maps may not be available or accessible. Therefore, we adopt a

navigation scenario in which blind users navigate based on route descriptions provided by sighted people, a method that is commonly used in practice [28, 4, 107]. In this scenario, it is unclear what kind of information users would need to enable blind people's decision-making in unfamiliar buildings. To build such a map-less navigation robot system in the abovementioned scenario, we aim to address the following research questions. (1) What kind of information is useful for blind people to reach a destination in unfamiliar buildings, given a route description by a sighted passerby? (2) Can we design a map-less navigation robot by using collaborative interaction inspired by guide dogs, and can blind users *decide* their way to their destination?

To answer these questions, we used a scenario-based participatory design approach [196] with five blind participants to understand what kind of environmental information would facilitate their navigation in unfamiliar buildings when using a navigation robot as their aid. During the study, an experimenter gave the participants a description of a route, which was gathered from an interview session with ten sighted passersby. Then, assuming the experimenter as a navigation robot, the experimenter accompanied the participants along the explained route while describing several indoor features along the way. Throughout the study, the blind participants mainly expressed that intersections and signs, such as directional signs (*i.e.*, signs which contain arrows to indicate where places are) and textual signs (*i.e.*, signs which only contain texts, such as room numbers and names of places), are the most useful information when navigating unfamiliar buildings.

Based on these findings, we designed and prototyped a map-less navigation system, called *PathFinder* (Figure 5.1), on top of a suitcase-shaped robot. Adopting a similar approach with guide dogs collaborative interaction, users can command the system via its handle interface to find the next intersection (Figure 5.1–C) or the end of a hallway, and describe visible directional and textual signs (Figure 5.1–D) to identify the path to the destination. The system adopts audio feedback to the user to convey detection results, such as the shapes of intersections and descriptions of signs.

A session for design iteration was conducted with the same five blind participants to gather feedback and comments about the interface and functionalities of the system. Through the study, we obtained suggestions regarding the system's audio feedback and handle interface. The participants also requested an additional "Take-me-back" functionality, where the system takes the user back to the location where they started their navigation.

Finally, we conducted a user study with seven blind participants on the system after incorporating the suggestions from the participatory study. During the study, we prepared two routes with several intersections and signs and asked the participants to navigate them using PathFinder and a topline system, which is a navigation system with prebuilt maps. Through our interview with the participants, we found that the participants felt they were able to navigate to the destination with increased confidence and less cognitive load with PathFinder compared to their daily navigation aid. In addition, while all participants mentioned that PathFinder required more effort for them to control than the system with prebuilt maps, they agreed that PathFinder is a useful navigation system as it can operate in more places, and they were able to navigate to their destinations without having to be accompanied by a sighted person.

Below, we summarize the contribution of this paper.

1. We propose a map-less navigation robot system for navigating blind people in unfamiliar buildings. To design the system, we performed a participatory

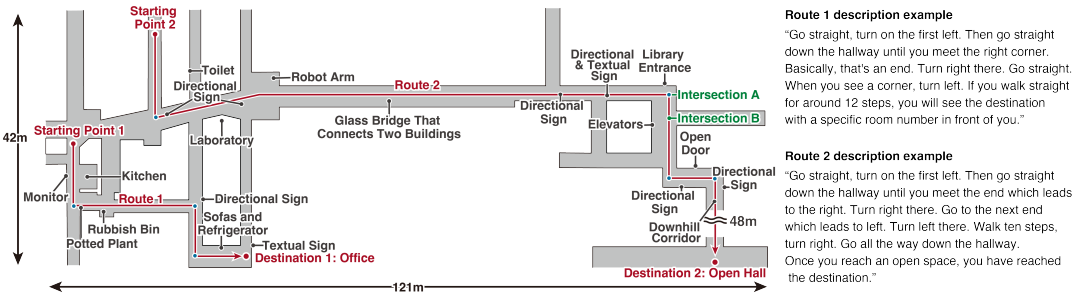


FIGURE 5.2: **Routes Used Throughout the Chapter.** A floor map of the building where the study was conducted, showing the two routes, various POIs along the routes, and examples of the route descriptions.

study with blind people to gather their insights and suggestions. Based on the study, we designed the system to recognize signs and detect intersections, then convey information to the blind user via audio feedback.

2. We conducted a quantitative and qualitative user study of the proposed map-less navigation robot system with seven blind participants. Based on the result, we discuss the functionalities and the limitations of the system, and also provide insights for designs of future map-less navigation systems.

5.2 Related Work

In this section, in addition to the related works reviewed in Chapter 2 — including navigation in unfamiliar buildings (Section 2.1.2) and map-based navigation assistance systems (Section 2.2.1) — we describe map-less navigation technology for robots, the shared control paradigm used in collaborative interaction, and the environmental information conveyed to blind users, such as intersections and signs. This builds on the literature reviewed in Section 2.3.1, Section 2.4.3, and Section 2.5.

5.2.1 Map-less Navigation Technology for Robots

In robotics, many works have studied the problem of navigation in environments without prebuilt maps. They mostly relied on vision-based techniques; for example, matching real-time RGB images with sequences of pre-captured images for path following [197, 75], recognizing the surrounding objects to help with localization [76, 75], and using an image of the target location and applying reinforcement learning to find the path to the goal [81, 82]. On the other hand, methods of navigation that create maps during exploration have also been proposed. Examples of these approaches include the construction of topological maps by detecting intersections [83] and occupancy maps using RGB images and deep reinforcement learning [84] while the robot is navigating the space. These methods do not require any information prior to the navigation because they aim to explore the environment. Inspired by these works, we utilize the navigation technique proposed for robot exploration to help navigate blind people in unfamiliar buildings, particularly the intersection detection technique.

Intersection detection algorithms have been used for robots to determine their path when maps are unavailable. In our case, using an intersection detection algorithm is necessary to imitate collaborative interactions with guide dogs and blind users. Garcia *et al.*proposed a method to detect indoor intersections only with RGB

images using a rule-based algorithm [148] and convolution neural networks [149] for quadcopters. Intersection detection in complex environments such as outdoor and underground mines has also been explored in past studies using a LiDAR sensor [198, 83, 199, 200] and an RGB camera [201]. In particular, Yang *et al.* [83] proposed a method to detect intersections in arbitrarily shaped environments using a 360° LiDAR sensor and a real-time SLAM algorithm. However, since their motivation is to explore novel environments quickly and create extensive LiDAR maps in a short time, the robot may travel in a manner where it does not take an accompanying blind person into account. For example, a robot may move very close to a wall or change its orientation frequently. To build a map-less navigation system for blind people, we apply the method of Yang *et al.* [83] to detect intersections, and include additional functionalities to take into account the accompanying blind user.

5.2.2 Shared Control for Robots

Without prebuilt maps and reference information of the destination, navigation systems cannot determine the path to reach the destination. To handle this issue, we employ shared control, *i.e.*, a method to control a robot using both human decision and the functionality of a system. According to Wang and Zhang [90], shared control is defined as a *"case in which the robot motion is determined by both the human operator and robot decisions in a mostly balanced fashion."* Shared control can be separated into near-operation, in which the operator perceives the scene with their direct sense, and teleoperation, in which the operator perceives the scene indirectly, such as through a screen. For example, near-operation has been used for assisting a driver to keep their vehicles in lane [91], controlling a wheelchair [92, 93, 94], and for assisting blind people to navigate in familiar buildings [95, 96, 97], while teleoperation has been used for navigating where a human cannot go [98, 99, 100], or for reconnaissance [101]. Similar to previous work, PathFinder adopts near-operation shared control, so that users can complement missing map information that comes from map-less restriction, while the system can help users to navigate safely through buildings. Furthermore, our proposed system allows users to effectively determine the way to the destination in unfamiliar buildings by conveying environmental information, which is described in the next section.

5.2.3 Conveying Environmental Information to Blind People

For the purpose of supporting blind people during navigation, researchers have worked on systems that can convey environmental information such as traffic lights [112, 113], doors [114, 110], intersections [22], and signs [115, 110, 116]. We particularly focus on conveying information about intersections and signs, as blind participants in our study have indicated that they are useful in unfamiliar buildings, which will be described in Section 5.3.2.

Intersection Information

Detecting and/or utilizing intersection information has also been done in the field of accessibility to provide turn-by-turn instructions to blind users [9, 22, 20, 95]. Lacey and MacNamara explored a smart walker with passive traction to navigate the elderly blind by informing intersections as landmarks in a controlled and familiar building, such as a residential home [95]. Also, Kuribayashi *et al.*proposed a system for blind people that conveys the location and shape of intersections by detecting

TABLE 5.1: **Top Three Information Described.** The table shows the top three pieces of information and the number of intersections described by sighted passersby.

Route	Top Three Information in Route Descriptions by Sighted Passersby		Number of Intersections Described	
	For Sighted Questioner	For Blind Questioner	For Sighted Questioner	For Blind Questioner
R1	Intersections with directions (70%) End of the hallway (40%) Doors along the way (40%)	Intersections with directions (100%) Distance to walk (60%) Where the wall are (60%)	Mean = 2.0 Median = 2.0	Mean = 2.4 Median = 3.0
R2	Intersections with directions (100%) Downhill corridor (60%) Existence of the library (50%)	Intersections with directions (100%) Downhill corridor (60%) Distance to walk (60%) Where the wall are (60%)	Mean = 2.9 Median = 3.0	Mean = 3.6 Median = 4.0

them using a LiDAR map constructed with a smartphone [22]. In their study, they revealed that conveying the shape of the intersection is effective for the navigation of blind people, as it helps them to localize themselves and to learn about the environment [22]. Therefore, based on their findings, we also convey the shape of the intersection every time blind users reach it.

Sign Information

Sign information has been considered a useful object to detect, as they generally contain information about the surroundings. In the area of computer vision, researchers have aimed to recognize texts on signs in the real-world [202] or detect sign boards [203]. On the other hand, signs that appear in indoor environments contain arrows that correspond with words that represent locations. Thus, to assist blind people to determine their way, a different system has been proposed in the field of accessibility. Saha *et al.* [110] developed a system that can read signs on a smartphone and revealed that reading textual signs (*e.g.*, names of surrounding shops) can help blind people reach their destination. Yamanaka *et al.* [116] proposed a method to recognize all directional signs using a 360° RGB camera and verified that their system helped blind participants make a decision at intersections in tactile pavings. However, each of them reads either directional signs or textual signs. In contrast, we propose a sign recognition algorithm that can distinguish and read both directional and textual signs. We achieve this by utilizing an object detection model to detect arrows and words, and by proposing a new algorithm that will analyze their correspondence.

5.3 System Design Based on the Preliminary Investigation with Sighted Passersby and Blind People

This section describes the system design of our system based on the investigation with sighted passersby and participatory design. To do so, we adopted a scenario-based approach to consider the design of the system [196]. The scenario used for our study sessions is as follows: *A blind person is navigating in an unfamiliar building with a navigation robot to reach his/her destination. As the building is unfamiliar to the blind person, he/she acquires the route description from sighted passersby in the building.* We prepared two routes with different characteristics for this scenario-based study. To make the scenario more specific, we first conducted an interview with ten sighted passersby to investigate what route description they would convey to blind people. Then, we had a design session with five blind participants to understand what information would be useful for them to navigate an unfamiliar place with only the route description from sighted passersby. This study was approved by the institutional

TABLE 5.2: **Blind Participant Demographic In Our Participatory Study.** The tables show their age, age of onset, gender, and navigation aid (P01–P05).

ID	Age	Age of onset	Gender	Navigation Aid
P01	74	32	Male	Cane
P02	67	10	Female	Guide dog
P03	38	0	Female	Guide dog
P04	76	0	Female	Cane
P05	59	0	Male	Guide dog

review board (IRB) of our institution, and an informed consent was obtained from every participant.

5.3.1 Routes For The Study

Route 1 (R1), shown on the left side of Figure 5.2, is a narrow corridor in a building. The route has three intersections (indicated in blue dots) and its length is approximately 46 m. It also has furniture, rubbish bins, a kitchen, and signs indicating room numbers along the way. Route 2 (R2) is a wide corridor that spans two buildings. The route has four intersections and its length is approximately 166 m. The route also has a glass bridge, a library, an elevator, and signs indicating the names of the buildings along the way.

Interview with Sighted Passersby

We recruited ten passersby who knew both R1 and R2 and conducted a ten-minute interview with $5 of compensation. For each route, we asked the participants to describe the route two times for two different cases: one for a sighted questioner, and another for a blind questioner. An example of a route description given by them is illustrated on the right side of Figure 5.2.

Table 5.1 shows the top three pieces of information and the number of intersections described by the participants. We found that descriptions of intersections are always mentioned when assuming the questioner is blind, but could be omitted when assuming the questioner is sighted. Also, when we counted the number of intersections described in the route descriptions, both the mean and median values were higher when assuming the questioner is blind. Participants described the difference in explaining the routes to sighted and blind people as follows. **C5.1**:*"It's not possible for blind people to read any graphical signs. For example, signs like the map of the building, plates on the wall which have room numbers, and signs hanging from the ceiling. Without any of these details, the best information I can convey is about which directions to take when it's needed."* These results indicate that sighted passersby describe which directions to turn at intersections particularly carefully to blind people.

5.3.2 Scenario-Based Study With Blind People

We recruited five blind participants (P01–P05 in Table 5.2). For each participant, we conducted a brief interview to gather their experience in navigating unfamiliar buildings. We then asked the blind participants to provide information that would make them confident in unfamiliar buildings if they are navigating with a navigation robot system. To do so, an experimenter first explained R1 and R2 with the description given by the sighted passersby in the previous interview (Section 5.3.1), and

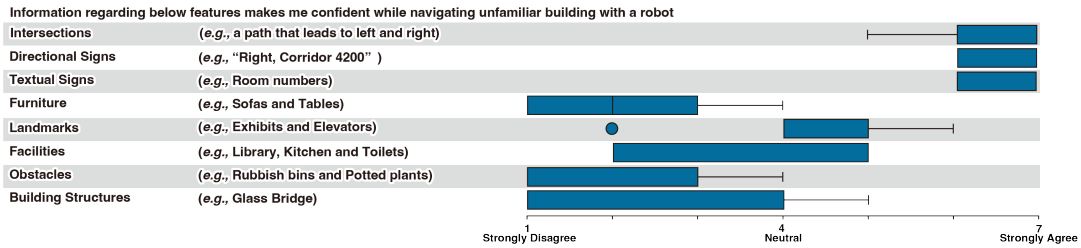


FIGURE 5.3: **Useful Indoor Features.** Questions and responses from our participatory study with five blind persons about useful indoor features. Responses are ratings on a seven-point Likert scale.

then guided blind participants along the routes, asking them to think of the experimenter as a robot. During guidance, the experimenter explained indoor features, such as intersections, directional signs, textual signs, furniture (*e.g.*, sofas, refrigerators, monitors, and tables), landmarks (*e.g.*, a robot arm and elevators), facilities (*e.g.*, libraries, kitchens, laboratories, and toilets), obstacles (*e.g.*, rubbish bins and potted plants), and building structures (*e.g.*, glass bridge and downhill corridor). After that, we asked the participants to rate each piece of information based on how confident it made them about their surroundings on a seven point Likert scale (1: Strongly disagree, 4: Neutral, and 7: Strongly agree). The interview took 75 minutes and each participant was compensated $35.

Results

All the participants answered that they have experience navigating unfamiliar buildings (*e.g.*, hospitals, airports, and universities) and agreed that they usually rely on sighted passersby. P04 mentioned their experience of navigating in unfamiliar buildings as follows. **C5.2**:*"When I'm navigating in an unfamiliar building, the only information I have is the room number (of the destination). I don't get any information about the route I have to take, so I have to rely on the first person I meet in the building to get there."* (P04)

Figure 5.3 shows the provided ratings for how useful each indoor feature was. The result shows that intersections, directional signs, and textual signs are relatively the most useful information when navigating an unfamiliar building with a robot. Taking this result into account, we designed the functionality of our proposed system as discussed in the next section.

5.3.3 System Design

Based on the interviews with both the sighted passersby and the blind participants, we designed Pathfinder to have two modules: An intersection detection module and a sign recognition module.

Intersection Detection

Based on our interview with sighted passersby, the route description conveyed to blind people mainly consists of which turns to take at intersections (Section 5.3.1). Blind participants also indicated that intersections are one of the most useful information in unfamiliar buildings. Therefore, we designed the system to be able to detect intersections together with their locations and shapes (*i.e.*, which way each intersection leads). Once the system detects an intersection, the system should convey

the intersection's shape to the user through audio feedback. By doing so, the blind user will be able to decide which way to go based on the original route information obtained from sighted passersby.

Sign Recognition

While intersection information may be sufficient to reach a destination, signs were found to make blind people more confident about their location. As indicated by the interview results (Section 5.3.2), the system should detect two types of signs, directional signs and textual signs. Directional signs are expected to help blind users confirm that they are on the correct route and help them make a decision at an intersection. Textual signs are expected to help blind users verify where they are and if they have reached their destination. As blind people cannot notice the existence of signs, the system should detect and notify the possible existence of a sign. Finally, as not all signs are relevant, the system should read out signs only if the user wants the system to.

5.4 Prototyping and Design Iteration

This section describes the initial prototype of our system and design iteration with the same five blind participants to improve our system.

5.4.1 Aapparatus

Employing Suitcase-shaped Robot

We adopted a suitcase-shaped robot called AI suitcase [17][1], as its appearance would allow the user to blend into a surrounding environment, such as a building in a metropolitan environment [61, 60, 11], and where the study was conducted in (Figure 5.1). AI suitcase, which was formerly named carry-on-robot (CaBot), will run alongside and slightly ahead of the user, enabling the system itself to be a protection by colliding with obstacles first [11]. Unlike quadruped robots that make a lot of walking noise [58], the form of a suitcase enables the system to take images from sensors stably with less motion blur [61, 59], which would allow the system to gain better recognition results of signs and intersections. While the weight of our current system is approximately 40 pounds, we expect that the size and weight of the computers and sensors in the suitcase to get lighter and smaller, which will enable users to carry the system around more easily.

Hardwares

The type used in this study was called the CaBot2-GT model[2]. The suitcase uses one that is on the public market, which is made by the GLOBE-TROTTER company[3]. Inside the suitcase, uses a mini PC with an NVIDIA RTX 3080 graphics board. For the environmental sensor, the robot uses a Velodyne VLP-16 LiDAR sensor and a RealSense camera on the top of the suitcase to sense surroundings (installed at about 0.7 m from the ground). RealSense camera could capture the RGB image with a resolution of 640x480. In the handle, there are four buttons placed left, right, front,

[1] https://github.com/CMU-cabot/cabot

[2] https://github.com/CMU-cabot/cabot_design/tree/master/cabot2-gt

[3] https://jp.globe-trotter.com/

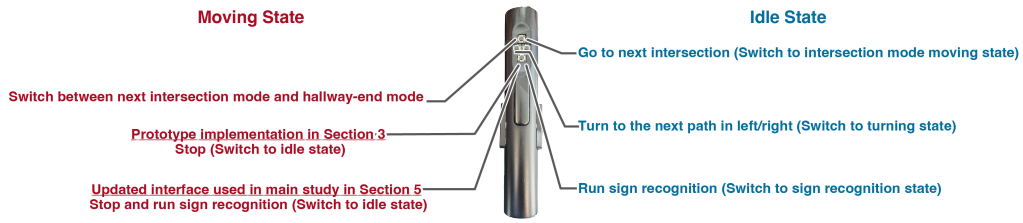


FIGURE 5.4: **Handle Interface of PathFinder.** It has four buttons, and each button is assigned different functionalities according to the system's state as indicated.

and back, with a touch sensor underneath the handle, which senses if the user is holding the handle.

Smartphone Interface

Also, a smartphone is paired with the suitcase for users to interact with the system. By using a dedicated application[4], users can either choose destinations via VoiceOver. Also, the audio feedback is provided through this application. The application and the PC are connected via Bluetooth to provide audio feedback and via Wi-Fi local network for fast information transmission.

Localization and Navigation

Existing AI Suitcase navigates users to their destination by using a predetermined map, using static route map, a LiDAR map, and BLE beacons. First, when the system is initiated, the system localizes itself by using BLE beacons and a LiDAR map, by using the Monte Carlo localization method. Once the users select their destination, the system plans a global path to the environment by using both the static route and the LiDAR map. While the navigation uses an RGB sensor and, LiDAR sensor to detect obstacles and their locations, along with a local path planning algorithm to ensure a safe path while navigating along the route. To ensure safety, the center of the robot is set to the user rather than the robot itself. Additionally, the robot only moves while the user is holding the handle, which is detected by the touch sensor of the robot.

5.4.2 Prototyping

We prototyped the first version of PathFinder based on the design derived from the first study session. The details of the implementation will be described in Section 5.5. Below, we first describe the handle interface and audio feedback of the prototype version of PathFinder.

Map-less Navigation States and Interface

The user can provide instructions to the system via the four buttons attached to the system's suit-case handle, consisting of a front button, a left button, a right button, and a back button (Figure 5.4). These buttons function differently when Pathfinder is in the idle state or the moving state, as described below.

When the system is in the idle state (*e.g.*, pausing at an intersection or at its initial position) pressing the left/right button will instruct the system to face the next

[4] https://github.com/CMU-cabot/cabot-ios-app

path which is on the left/right of the current facing direction while saying *"Turning left/right."* Pressing the front button will switch the system to the moving state and instruct it to move to the next intersection, saying *"Going to the next intersection."* Finally, pressing the back button will initiate Pathfinder's sign recognition module while also saying *"Recognizing signs."* Then, recognized signs will be read out after it finishes processing. An example of the audio feedback when three signs are recognized is as follows: *"There are three signs. 1. Forward, main lobby, 2. Right, Mechanical Engineering, and 3. Entering [proper noun] Hall."*

In contrast, if the system is in the moving state, only two buttons, the front and the back buttons can be used. Pressing the front button will make the system switch between the *Next Intersection* mode and the *Hallway-end* mode. In the next intersection mode, the system will navigate until it reaches the next intersection, where it would stop and convey the intersection's shape. We adopted the clock position to convey the shape of intersections as it is capable of conveying non-perpendicular turns, and can be easily generalized in public buildings. An example of the feedback when an intersection that leads to forward, left, and right is found is as follows: *"Found route to forward, two o'clock, and nine o'clock."* Meanwhile, the hallway-end mode instructs the system to move forward until it reaches the end of the hallway, ignoring all intersections along the path. This is useful when there are a lot of intersections and the blind user knows they will not be making any turns. Note that when this button is pressed, the system will say, *"Going to the next intersection/end."* On the other hand, pressing the back button causes the system to stop and switch back to an idle state while also saying *"Stop."* For simplicity of design, the system does not take any input while it is turning until the turn is done.

5.4.3 Design Iteration

After implementing the system, we conducted another session with the same group of five blind participants (P01–P05 in Table 5.2). The aim of the study was to improve the interface and functionality of the system. For each participant, we first introduced the system and asked them to use the system while walking along R1 and R2. We then interviewed the participants about areas for improvement. The interview took 75 minutes and each participant was compensated $35.

All five participants generally agreed that the prototype version of the system will be helpful when navigating an unfamiliar building. Still, we received comments to improve the system when we asked for suggestions. Below, we list the major suggestions obtained from participants and a summary of updates made to the interface of the system.

Intersection shape should be conveyed using "left, right, forward, backward" terminology

Three participants mentioned that intersection shape should be conveyed with left and right, and not with clock position. As the two routes in the study contained only perpendicular intersections, we updated the audio feedback so that the system conveyed the intersection shapes using "left, right, forward, backward" terminology.

Position of textual signs should be conveyed, and fewer signs should be read

P04 pointed out that the system should convey the position of textual signs. He indicated that conveying its position is important, so that he knows where exactly the

destination is if the system reads out room numbers. As such a feature is important for the last-few-meters problem [110], we updated the system to read out the distance and position of textual signs. Specifically, the system will convey the direction (*e.g.*, left, left wall, right, right wall, and front) and distance to a textual sign.

In addition, three participants mentioned that the amount of information from signs was overwhelming as PathFinder read out all signs in its field of view. In the study, there were several situations where it read out more than six signs. Therefore, we updated the system so that it will read a maximum of four signs. The system will first read directional signs, then textual signs if directions in the directional signs were fewer than four. When reading textual signs, the room number is read preferentially. Overall, we updated the audio feedback as follows: *"There are two directional signs. Left, corridor 4600, and right, (corridor) 4508 to 4533. Also, there is one textual sign saying room number 4521 to your front, 2.1 m ahead."*.

Associating information from directional signs with the turn direction at intersections

In the prototype version of PathFinder, the system only read out directional signs when the user initiates sign recognition. However, P01 pointed out that it would be helpful if the system could also read out where the system is turning to, from a nearby directional sign when a turn is being made at an intersection. For example, if a sign with "Right, corridor 4200" is recognized near an intersection that leads to the right and the user instructs the system to turn right, the system should say, *"Turning to the direction of Corridor 4200"* when making the turn. We implemented this feature on the system as it may increase its usability.

Merge stop button and sign recognition button

Three participants stated that the current interface of needing to press the back button twice to recognize a sign from the moving state is a cumbersome process. Hence, we updated the button layout of the handle interface so that sign recognition can be initiated even if the system is moving. Pressing the back button causes the system to switch back to the idle state and instructs the system to run the sign recognition algorithm, saying *"Recognizing sign."*

Add "Take-me-back" functionality

P01 pointed out that it would be helpful if the system could take them back to the initial position where they started from, as such a task is difficult for blind people [204]. Therefore, we added a "Take-me-back" functionality to the system. As the system constructs a cost map and accumulates the map information over time, the system is able to maintain information about the initial position and return to it.

5.5 Implementation

This section describes our implementation of PathFinder based on the design considerations in the previous section. PathFinder requires a 360° LiDAR and an IMU sensor to construct a LiDAR map on the fly and detect intersections. It also requires a camera that can capture indoor signs as clearly as possible. We chose the open source robot platform CaBot[5] as our base platform and extended it for PathFinder.

[5] https://github.com/CMU-cabot/cabot

To navigate in a building without prebuilt maps, our system constructs a LiDAR map of its surrounding environment in real-time by using Cartographer[6], an open-source SLAM implementation. Our system operates an algorithm on this real-time map to find the paths that the system can navigate, and informs the user about intersections.

Although the system has an RGBD camera, we attached a smartphone with a high resolution camera (iPhone 12 Pro) on an extendable stabilizer (Figure 5.1–A, at 1.1 m from the ground) so that it could effectively capture indoor signs from a higher position and with a higher resolution compared to the RGBD camera. Here, the iPhone was chosen to enable fast prototyping of the system. Future versions may instead use an extra camera in an integrated manner. Below, we describe the details of our intersection detection and sign recognition algorithms.

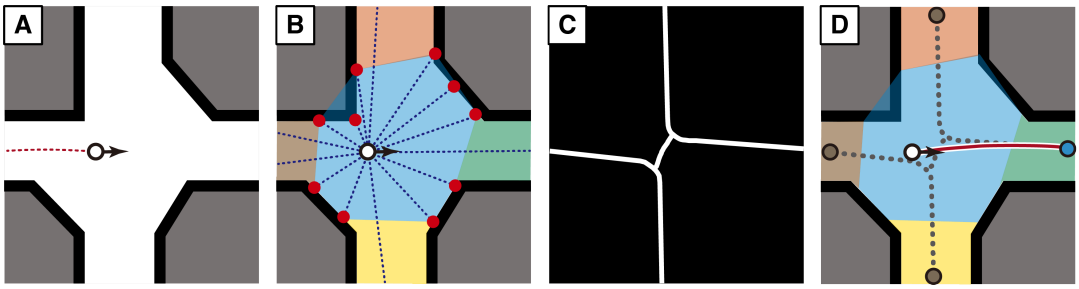


FIGURE 5.5: **Intersection Detection Algorithm Steps.** (A) Raw LiDAR map created by the system using SLAM (B) Detecting the closest convex hull similar to the method of Yang *et al.*[83] (C) Skeletonizing the regions outside of the convex hull to identify waypoints (D) Detected waypoints in the corridor regions, with the waypoint selected by the user shown in blue.

5.5.1 Intersection Detection

To detect intersections, we use the method proposed by Yang *et al.* [83]. The system first extracts waypoints for where the system can move from the latest LiDAR map. If there are waypoints found on the left or the right sides of the system, this means that the system has detected an intersection. In such a case, the system will stop and inform the user of the detected directions so that the user can determine which way to turn. In addition, the algorithm is further used to make the system move in the middle of the corridor for their safety [22], and keep the heading direction of the system facing the corridor.

Our overall algorithm to extract waypoints runs at about 10 Hz, and can be described as follows.

1. The system extracts a region of size 20 m $\times$ 20 m around the system from the latest LiDAR map (Figure 5.5–A).

2. Based on the extracted map, the system applies the following steps proposed by Yang *et al.* [83] to detect all the corridors leading to the intersection: (Figure 5.5–B).

   (a) The system samples points starting from the surrounding obstacles (*e.g.*, walls) within a certain radius (8 m) of the system, at constant angular intervals (10°), as shown with red circles.

[6] https://github.com/cartographer-project/cartographer_ros

(b) The system computes a convex hull using the sampled obstacle points, shown as the blue region.

(c) The system extracts the obstacle-free areas from outside the convex hull as the corridor region(s) (colored regions outside the convex hull).

3. The system extracts the topology of the middle of the detected corridors by skeletonizing the image of the LiDAR map (Figure 5.5–C).

4. Finally, the system assigns the furthest point on the topology from the system in each corridor region as waypoints that the user can instruct the system to move to (Figure 5.5–D).

Note that if the system does not detect any corridor on either side of the system, then the system will continue to move forward until it is stopped by the user or until it finds an intersection or the end of the hallway.

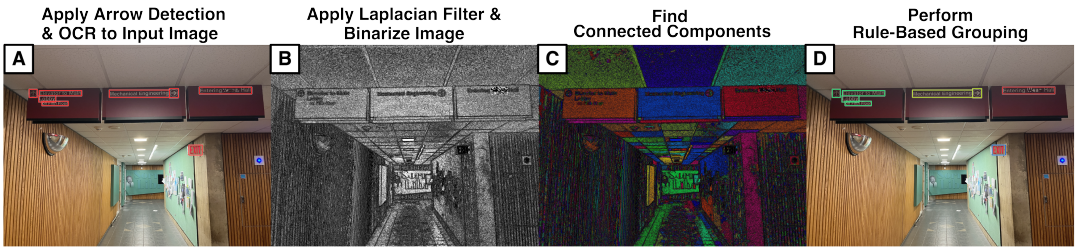


FIGURE 5.6: **Sign Recognition Algorithm Steps.** The system first applies arrow detection and OCR (A), followed by Laplacian filtering and image binarization (B). Then the system finds the individual regions in the image using connected component analysis (C), followed by a rule-based grouping to associate the arrows and recognized text (D).

5.5.2 Sign Recognition

Indoor signs are often placed on the ceiling or on the walls at a distance, and they appear very small in the images. To recognize texts and arrows accurately, we need an OCR model and arrow detection model which require about 5 seconds on average to process on our PC (including communication times), which may be too long for a blind user to wait for [110]. Therefore, we implemented a sign detection module. The sign detection module informs the user about the possible existence of a sign in real-time, without fully recognizing the sign, and the user can initiate sign recognition if they think the sign may benefit them to reach the destination. Here, we describe the sign detection module and the sign recognition module.

Sign Detection Module

The module is implemented as an iPhone app by using the Optical Character Recognition (OCR) model from the iOS Vision API[7]. The model detects texts approximately at five Hz on the iPhone, and the depths of the detected texts are then measured using its LiDAR sensor. If a piece of text is detected in five consecutive frames and is found to be within 6.5 m, the system notifies the user of the possible existence of a sign by saying, *"There might be a sign."* If the user then chooses to initiate the full sign recognition, the system will then send the image taken by the smartphone to the PC on the system via the local network. The system will not inform the existence of a sign which is within 5 m from the previously detected sign.

[7] https://developer.apple.com/documentation/vision/recognizing_text_in_images/

Sign Recognition Module

When the user initiates the sign recognition by pushing the back button, the system commands the iPhone to send an RGB and depth image to the sign recognition server running on the system's PC via the local Wi-Fi network. Once the image is sent to the server, the system runs OCR using the EasyOCR Python library[8] which is more accurate than the iOS OCR, and runs YOLOv5 [205] object detection model to detect the arrows in the image. We trained the object detection model to detect and classify eight categories of arrows (four horizontal and vertical directional arrows, and four diagonal directional arrows). An example illustration of the detection result is shown in Figure 5.6–A. According to the system design (Section 5.3.3), it is necessary for the system to recognize both directional and textual signs. To do this, the system needs to associate the detected arrows with the detected texts to recognize a directional sign, while also separating the texts from the arrows if there is a textual sign. To do this, the system assumes texts and arrows can be grouped together if they have a similar background color. Below, we describe the steps of our grouping algorithm for signs.

1. The system first detects edges in order to separate regions with different background colors. This is done by applying a Laplacian filter to each of the RGB channels, then for each pixel, selecting the highest value over the channels to create a single-channel image. This will result in an image where pixels with high values represent edges, while those with low values represent non-edges.
2. The system then obtains a binarized image by assigning 0 to pixels whose values are higher than a pre-determined threshold value, and 1 to all other pixels (Figure 5.6–B).
3. The system applies the connected component labeling algorithm to the binarized image to determine the regions with similar background colors, and obtain a region label for each pixel (Figure 5.6–C).
4. The arrows and texts whose bounding boxes have the same region label are then grouped together (Figure 5.6–D).

As a result, the grouped sets of arrows and texts will be obtained. Note that a text may not necessarily have to be grouped with arrows, as the algorithm only considers the background color of each bounding box. Note that in practice, a grouped set may include multiple arrows and texts. Deciphering matches between multiple arrows and texts would involve a more complicated algorithm outside of the scope of this study. So we only consider signs where each text corresponds to only one arrow. The system calculates the euclidean distance between the center of the bounding boxes of arrows and texts, and groups texts with arrows that have the smallest distance between them. Finally, using the LiDAR sensor of the iPhone 12 Pro, the system removes signs that are further than 6.5 m from the recognition results, so that only signs with accurate detection results are conveyed to the user.

5.6 Main Study

The main goal of this study is to understand the effectiveness of our complete system and how well it can assist blind people. For comparison, we used a system with prebuilt maps as the topline reference, which we assumed to provide the best possible

[8] https://github.com/JaidedAI/EasyOCR

TABLE 5.3: **Demographic of Blind Participants in the Main Study.** Additionally listed are the main study participants' normalized task completion times, and the number of times they asked for the route during navigation. *1: normalized as they chose a slower speed. *2: normalized as the previous user's topline setting was used accidentally

ID	Age	Age of onset	Gender	Navigation Aid	SUS	Normalized task completion time [sec]				Times asked for route	
						Route 1		Route 2		Route 1	Route 2
						PathFinder	Topline	PathFinder	Topline	PathFinder	
P06	68	0	Male	Cane	87.5	398.3	59.2	491.5	240.7	4	6
P07	69	49	Female	Cane	95.0	174.5*1	66.3*1	919.8*1	234.7*1	0	3
P08	63	0	Female	Cane	72.5	854.3*1	66.1*1	914.9*1	239.0*1	2	6
P09	63	56	Male	Cane	80.0	260.4	66.7*2	601.5	230.8	0	3
P10	74	0	Female	Cane	92.5	223.6	60.1	404.2	259.9	0	2
P11	63	3	Female	Cane	90.0	147.3	60.9	350.3	240.6	0	3
P12	50	1	Male	Guide dog	52.5	163.1	63.0	573.0	232.1	3	2
Mean	64.29				85.25	317.36	63.16	607.88	239.69	1.29	3.43
±SD	±7.52				±15.74	±251.68	±3.20	±228.85	±9.78	±1.70	±1.80

navigation experience. We conducted the main study with seven blind participants (P06–P12 in Table 5.3). This study was approved by the IRB of our institution, and an informed consent was obtained from every participant. Each study took 150 minutes and participants were compensated $70.

5.6.1 Tasks and Conditions

We asked the participants to navigate R1 and R2 (Section 5.3.1) from the starting points to the destinations. We prepared two conditions, one where participants used PathFinder and the other where they used the topline system *i.e.*, a system that uses prebuilt maps. We did not include the condition of using only the regular aid for safety concerns, as blind people usually do not navigate in unfamiliar buildings without an assistant [4, 28].

PathFinder

For the condition using PathFinder, we first described the route to the destination to the participants, then asked them to navigate to the destination. They were allowed to ask the experimenter for the route during the task if they needed it, in which case the experimenter would describe the route again from their current position to the destination. The number of times they asked for the route description is reported in Table 5.3. The experimenter intervened in the study only if the participants turned at the wrong intersection or the system malfunctioned.

Topline System

For the topline system, we used the original AI suitcase [17][9] that can navigate blind people in buildings with prebuilt maps. To operate the topline system, we constructed full prebuilt maps of R1 and R2, which are annotated with POIs such as intersections, building names, and facility names. When the system is initiated, it localizes itself in the prebuilt map using BLE beacons, which are attached to the building. In the study, the experimenter manually set the destination remotely via a smartphone application, and the system started navigating to it once the user pressed a button on the handle. During navigation, the system reads out annotated POIs along the way, and while turning the vibrator which is in the direction of

[9] https://github.com/CMU-cabot/cabot

the turn vibrates. When the navigation ends, the topline system indicates that they have arrived at the destination. Note that the duration to navigate through R1 and R2 does not depend on a participant, but mostly on the social context of the building (*e.g.*, a crowd of people), as the topline system automatically navigates to the destination at a constant speed.

5.6.2 Procedure

We first introduced PathFinder and explained its map-less navigation feature and conducted a 30-minute training session. In the session, we adjusted the speed of the system to 0.75 or 0.50 meters per second (m/s) based on each participant's walking speed. The adjusted speed was used for all tasks and scenarios for the participant. Then, the participants were asked to navigate through R1 and R2 with PathFinder based on the route descriptions we gave them. At the end of each route, they were also asked to use the "Take-me-back" functionality and go back to the initial position. Next, the participants were asked to navigate R1 and R2 again using the topline system. We did not counter-balance the order of the PathFinder and the topline systems because we did not want to induce any route learning in the user by using the topline system prior to using PathFinder. If we counter-balance the order of conditions and let several participants perform the topline system first, the task using PathFinder will be significantly easier due to the prior knowledge of the route walked with the topline system, as PathFinder will require users to memorize the route description while topline system does not. The tasks were video recorded so that it allows us to complement quantitative results. After the navigation, we asked the participants to answer a set of questions (Q1–8 in Figure 5.7) on a seven-point Likert scale (1: Strongly disagree, 4: Neutral, and 7: Strongly agree). We asked Q1–Q3 three times, to compare their experience when using their regular aid (*i.e.*, canes or guide dogs), PathFinder or the topline system. Then we asked Q4–8 regarding the usability of PathFinder. Finally, we asked participants to rate PathFinder using system usability scores (SUS) [174] and gathered open-ended questions for qualitative feedback.

5.6.3 Metrics

We used three metrics to evaluate and analyze the usage of the proposed system.

Task Success Rate

We define task success as an occasion where users successfully arrive at the destination without any intervention by the experimenter, caused by significant confusion during the experiment, such as turning in the wrong direction or walking past an intersection that had to be turned.

Normalized Task Completion Time

We measured the time taken to complete each task. We started the timer when the participants pressed the button to initiate the system at the starting point. The timer was stopped when the participants verbally indicated that they had arrived at the destination and were 5 m within the destination. As some of the participants used a slower speed (0.5 m/s), we normalized their task completion times such that all times correspond to 0.75 m/s. This is calculated as $T_m \times \frac{0.5}{0.75} + T_s$ where T_m is the

total duration the user and the system are moving together, and T_s is the duration for which they are standing still (which is thought of as the users' decision making time).

Performance of Intersection Detection and Sign Recognition

To evaluate the performance of the system, we measured two metrics based on the logs recorded by the system during the study. For intersection detection, we classified a detection result into four cases: (1) *correct detection* when the system correctly detected the intersection's shape, (2) *partially correct detection* when the system detected the turn direction, but missed some directions which are not a direction of turn, (3) *failed detection* when it did not detect the turn direction, and (4) *false positive detection* when the system detected an intersection where there was none (*i.e.*, at straight corridor).

For sign recognition, we classified detection results into four cases: (1) *correct and relevant recognition* where the detection result contained information to reach the destination, (2) *correct but irrelevant recognition* where the result was correct but contained only irrelevant information, (3) *null recognition* where the system did not recognize any sign, and (4) *wrong recognition* where the system was unable to recognize sign correctly as either arrow detection, OCR or the grouping algorithm failed. Note that the number of correct and useful recognitions and correct recognitions may vary depending on the route and destination.

5.7 Results

5.7.1 Overall Performance

Task Success Rate

Out of 14 trials, ten were successful and four required intervention by the experimenter, resulting in a 71% success rate. For both R1 and R2, the task success rate was 71%, with two failures and five successes each. Failures occurred when users turned at the wrong intersection or walked past an intersection where a turn was required, as will be described in the next section. This outcome arose despite allowing participants to ask for the route, indicating that although they expressed confidence in their navigation, they chose incorrect directions. In other words, their perception of the route did not align with the actual environment.

Normalized Task Completion Time

Table 5.3 shows the results of the task completion time. Statistical analysis using the Wilcoxon signed-rank test revealed that tasks with PathFinder took significantly longer to complete compared to those with the topline system ($p < .05$ for both R1 and R2). Below, we summarize four reasons why our proposed system took a longer time to complete a task. 1) Our system took extra time because PathFinder stopped at each intersection. 2) Participants took time to recall and determine the direction to turn. 3) Our system required time to run sign recognition each time it was initiated. 4) There were four instances in which participants turned at the wrong intersection (occurring twice for P08 and once for P06, P07, and P09), requiring additional time for them to notice the error and return to the correct route with intervention by the experimenter. 5) P08 was particularly confused about the interface of the system and took extra time to complete the tasks.

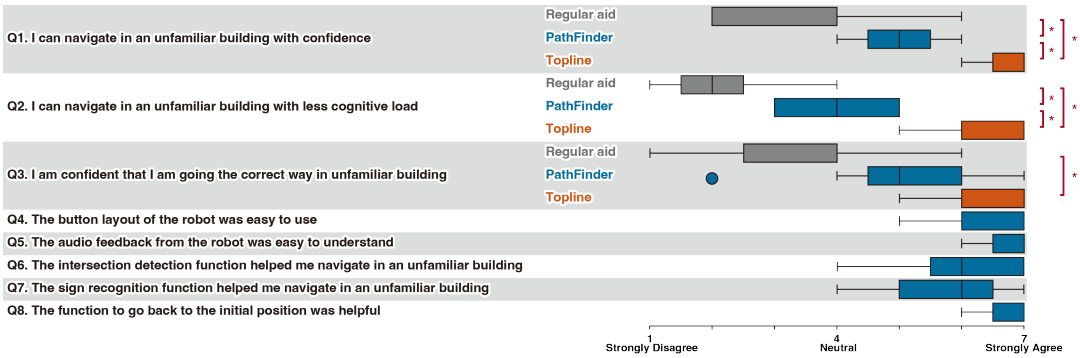


FIGURE 5.7: **PathFinder Was Rated Between Regular Aid and Topline System.** Questions and responses from our main study with seven blind participants. Responses are rated on a seven-point Likert scale. Responses marked with * indicate a significant difference between the systems when applying the Wilcoxon signed-rank test (p<.05).

Subjective Ratings

Figure 5.7 shows the results for Q1–8. Statistical analysis using the Wilcoxon signed-rank test revealed that participants felt that PathFinder is significantly better compared to their regular aids for Q1 and Q2 ($p < .05$). The same test also revealed that the topline system received significantly better ratings than the proposed system for Q1 and Q2 ($p < .05$). For Q3, there was no significant difference neither between PathFinder and regular navigation aids ($p = .14$) nor between PathFinder and the topline system ($p = .09$). Finally, Table 5.3 shows the SUS scores given by each participant.

5.7.2 Performance of Intersection Detection and Sign Recognition

PathFinder detected intersections 108 times in total throughout the whole study. There were 61 correct detections, nine partially correct detections, five failed detections, and 33 false positive detections. The partially correct and false positive detections did not harm the performance of the task, as it still detected the way the blind participants have to go. The false positive detections mainly occurred when navigating through a glass bridge in R2. The system detected false intersections on this straight bridge, as glasses are transparent from the LiDAR sensor. Two failed detections out of five occurred at intersection A and intersection B (Figure 5.2) when P07 was navigating R2 in a significantly crowded situation. In intersection A, the path was crowded in front of the elevators in R2, and the system did not detect the path that leads to the right. Also, the system detected intersection B, which leads to the front and left, as an intersection that leads to left and right, as the crowd affected the orientation of the system to face in between the two paths. The other three failed detections were related to system errors.

Participants initiated sign recognition 62 times in total while navigating R1 and R2. Throughout the whole study, P06–12 initiated sign recognition 1, 16, 22, 10, 4, 4, and 5 times, respectively. There were 27 correct and relevant recognitions, 12 correct but irrelevant recognitions, 13 null recognitions, and ten wrong recognitions. Null recognition occurred when the participant initiated sign recognition where there was no sign. They initiated the sign recognition as the sign recognition module notified them of the existence of signs, but by the time they press the back button, the sign was out of the camera's field of view. Also, participants who took the wrong path

in their tasks tend to use the sign recognition function much more times, even when the system did not notify the possible existence of signs (P07 and P08).

5.7.3 Video Observation

Confusion When Using Sign Recognition

During the study, we observed an occurrence where the interface of the PathFinder confused two participants. P08 and P09 performed sign recognition on a directional sign (*e.g.*, "Left, Corridor 4200, and right, Corridor 4100") at a spot before the desired intersection (about 1-2 m). After listening to the feedback, they immediately pressed the left/right button as they thought it would take them to the next path. As the system was before the intersection and had not detected it yet, the system turned backward. While P09 was able to recover after a short time of confusion, P08 was quite confused with the occurrence, because the system did not move as she wanted it to. **C5.3**:*"It confused me when it was telling me that a sign was available, but I am not at an intersection... like when it was telling me to take the right turn to Destination 2, but I went down the wrong corridor because I wasn't yet at the intersection to make the turn."* (P09)

Navigation Error Occurred When Intersection Detection Failed

The intersection detection error in intersections A and B (Figure 5.2) both occurred when P07 was performing the task. In intersection A, the system was able to detect the shape of the intersection correctly when the system ran the intersection detection algorithm again after the crowd was eased. When the detection failure in intersection B happened, she instructed the system to turn in the wrong direction instead of going forward. She realized that she had turned in the wrong intersection after reaching the dead end of the hallway and managed to recover back to intersection B.

5.7.4 Qualitative Feedback

Positive Feedback

Through the study, we received many comments indicating that they found the functions of PathFinder useful. All participants found the intersection detection feature of the system helpful, as it can find an intersection more accurately and quickly compared to their usual navigation aid. **C5.4**:*"(To find an intersection) I would have to stay close to the walls and feel with my cane, which is not always possible. But with the robot, I can quickly find the intersection without being close to a wall."* (P10)

In addition, the sign recognition feature of the system was generally appreciated by the participants. Participants described the advantage of textual signs and directional signs as follows: **C5.5**:*" I really liked the sign recognition. If I'm in an unfamiliar place like an airport, and it read out the gate number (textual sign), I would immediately know where I am and that there are more gates"* (P06) and, **C5.6**:*" Yes (directional signs are useful), it gives confirmation which is important, and you can get confidence about where you are going."* (P11)

Also, the "Take-me-back" functionality was appreciated by all seven participants from the main study. **C5.7**:*" It was very useful and felt almost like the with-map robot. It would be very convenient when I'm in an office building and want to quickly find my way back to the entrance. I can also envision the ability to key in my favorite spots while I'm*

exploring and then trust the robot to directly take me to those spots while I'm navigating the second time." (P09)

When we asked whether PathFinder is acceptable compared to the topline system, all participants agreed and were appreciative of its advantage that it could be used in more places compared to the topline system: **C5.8**:*" If a map is available, it is definitely the most useful. But maps are not available everywhere. However, even without map information, the robot is very useful because it reads out information around me and that lets me find my way."* (P07) While the topline system generally received higher ratings in Q1–3, two participants still found the controllability of PathFinder to be better than the topline system. **C5.9**:*" With PathFinder I felt like I was in more control as I had the ability to get feedback from it at every intersection and use my own judgment. But with the topline system, I just had to let it do its thing, which I'm not fully trusting of. "* (P09)

Negative Feedback

Participants were confused when the system indicated the possibility of the existence of a sign, without indicating what types of signs they were: **C5.10**:*" I felt like the system wasn't picking up all the signs, and when it did, it didn't say what kind of sign it was right away. It is important for some kind of signs to be known right away, like fire exits, instead of having to ask for it each time."* (P07)

P12 rated the SUS score the lowest. When we asked him for the reason, he commented that guide dogs would be a better tool for their secondary travel, as follows: **C5.11**:*" I think once I learn the layout of a building, I will be able to navigate much faster with my guide dog than with the robot."* (P12)

Comparison with Guide Dogs

When we asked P12, who is a guide dog user, about the difference between the proposed system and the guide dog, he mentioned two points, that guide dogs may miss an intersection unless the user is aware of it, and guide dogs do not remember how to return to previous places, as follows: **C5.12**:*" With guide dogs, I have to know when to turn. But the robot will tell me, so I won't miss the intersection... The guide dog doesn't always know how to go back. I have to remember the route and teach the dog. With the robot, I would just have to hit the button, and it would take me back. I didn't have to remember the route by myself."* (P12)

5.8 Discussion

5.8.1 Were Users Able to Navigate with PathFinder?

The overall results showed that users were generally able to navigate to their destination (success rate of 71%) using PathFinder. This shows that users can independently *decide* and navigate their way to their destination in an unfamiliar building by having route descriptions from sighted passersby, sign information, and intersection detection functionality, coupled with guide dog collaborative interaction. While the task completion time for PathFinder was significantly longer than the topline system, this was necessary for each participant to stop at each intersection to choose a direction. Still, participants appreciated the potential of the system to be applied to various locations. Although they asked for routes several times during the study, which showed a tendency for more frequent requests on longer routes, a potential

solution would include recording the route description or even using VLN technology to interpret the route description instead of the user, which opens a new approach for this system. Overall, the results suggest the potential of a map-less navigation system to be applied in various environments.

5.8.2 Comparison with Topline and Regular Aids

PathFinder may be considered to be an option that could be in between the topline system and regular aid for its performance, functionality, and usable area. The confidence scores were significantly higher than the regular aids, and lower than the topline system (Figure 5.7–Q1). The ratings for cognitive load were significantly higher than the regular aids (lower cognitive load), but lower than the topline system (higher cognitive load) (Q2). As for the usable area of the aid, regular aids are the largest, as there are routes in which PathFinder cannot be used, such as routes with steps or rough surfaces. The topline system has the smallest usable area, given the requirements of the prebuilt maps. Also, the topline system can announce POIs, and PathFinder can do this partially using its sign recognition module. The regular aids do not have the capability to announce POIs.

In short, PathFinder can be a unique "in-between" option for blind people. The topline system can provide highly reliable and robust navigation while announcing POIs, but its usable area is small. Regular aids, on the other hand, can be used in most environments, but their reliability is low. PathFinder's approach can realize novel scenarios such as visiting an unfamiliar indoor public space and navigate without prior preparation and sighted companion. We still have a long way towards the goal, especially for the recognition capabilities, but we believe this study showed a new possible solution.

5.8.3 Usability

The system received favorable ratings with medians of 6 and 7 on the Likert scale for usability scores (Q4: 6, Q5: 7, Q6: 6, Q7: 6, Q8: 7). The ratings for the interface show the success of our participatory design process. Also, the median ratings for intersection and sign detection were both 6. One of the reasons why neither of the scores was 7 may be the interface design related to sign recognition. PathFinder requires the participants to select directions via buttons after reaching an intersection, not after the announcement of directional signs (Section 5.7.3, and C5.3). P08 was particularly confused with the occurrence of accidentally turning backward after listening to information about directional signs that the system recognized (Section 5.7.3), and rated the score for Q3 (confidence in the walking direction) with a 4 for the regular aids and a 2 for PathFinder. Although all other participants rated PathFinder to be higher than the regular aids for Q3, there was no significant difference between PathFinder and the regular aids. Also, there was no significant difference between PathFinder and the topline system for Q3 as P09 rated PathFinder higher (PathFinder: 6, Topline: 5) and P12 rated both the same (both 7). To prevent the occurrence of accidentally turning backward, the system can ask the user for verification when turning before an intersection so that it can prevent the unnecessary backward turn (*e.g.*, *"You are trying to turn before an intersection. Are you sure you want to make a turn?"*).

5.8.4 Benefit of Collaborative Interaction: Contorallability

PathFinder adopted collaborative interaction, in which the robot takes over the mobility, and the user decides which way to go based on the robot's verbal feedback. Two participants indicated that PathFinder was even better than the topline system in terms of the trust, as they gained more controllability with the system (C5.9). They preferred a more controllable system in spite of the time disadvantage. The current navigation robot systems are designed by putting weight on automation more than controllability. It may be possible to integrate some controllability into the topline system, such as a route-choice interface at an important intersection toward a destination. At this moment, it is not clear how we can balance automation and controllability, but future navigation robot systems for blind people may wish to consider this as part of the design.

5.8.5 "Take-me-back" Functionality and Gradual Map Creation

All participants unanimously agreed that the "Take-me-back" function was useful, as returning to the entrance is generally challenging for blind people, especially in unfamiliar buildings. P12 indicated that guide dogs could not complete such a task, as guide dogs do not remember an unfamiliar building layout (C5.12). This feature also leads toward the discussion of the gradual creation of maps equivalent to prebuilt maps, by blind users themselves. It may accumulate the necessary data to enable with-map navigation for routes to typical destinations at the building if the system preserves constructed LiDAR map data and provides an interface for blind people to annotate the LiDAR map with visited routes and POIs (C5.7). Such a feature can be a game changer for navigation systems by paving the way toward city-wide maps.

5.8.6 Possible Improvements for PathFinder

The current intersection detection algorithm works robustly only on limited situations and buildings. For example, in the main study, PathFinder sometimes misinterpreted the shape of an intersection (Section 5.7.2). The system detected an intersection in R2 as a dead end, because the corridor was packed with a crowd of people. In such situations, it may be necessary for the system to determine the level of congestion, and emit a path-clearing sound, so that people may move out of the way for the system to eventually find an intersection [61]. In addition, the system made multiple false positive detections at the glass bridge in R2, as the system failed to extract the shape of the bridge to a LiDAR map because of the transparency of glass to LiDAR. Besides, the system can not detect intersections at buildings with open spaces such as open halls or wide corridors, such as an atrium, because the current implementation has an assumption of the size of an intersection (Section 5.5.1). It is also difficult for PathFinder to define a local destination in an open space, resulting in the system to go in a random direction, as the topology extraction is likely to fail. To increase the generalizability of PathFinder, it is necessary to consider various environments (*e.g.*, open hall, hallway with open doors, glass bridge, and a wide corridor) to improve the intersection detection algorithm, for example by also using RGB camera as well as LiDAR sensor.

As for the sign recognition feature, while it was appreciated by the participants (C5.5 and C5.6), the function that conveyed the existence of the signs was insufficient (Section 5.7.2). As P07 pointed out, the system needs to detect types (*i.e.*, whether it

is directional or textual) or relevancy (*i.e.*, whether it contains necessary information) of signs before notifying the user and running the full sign recognition (C5.10). As determining the types of signs will need to go through a sign recognition which takes 5 seconds to process, one solution is to determine the relevancy of a sign by using the information which the user can input prior to their travel [116]. The user may input some keywords, such as the name of the destination or room number, so that the relevancy can be determined by only the result of OCR.

5.8.7 Limitations

Limitation of Study Design

This study design had several limitations. Firstly, the study was conducted on only two routes in our institution's buildings. The performance of intersection detection and sign recognition may vary in other environments, therefore, a comprehensive evaluation of those functionalities is one of the important future works. We also made some assumptions with our choice of the study's environment to prototype the PathFinder system, such as routes without any steps and floor transitions. Features such as elevators or stairs may have been rated higher in the participatory study if the route contained floor transitions. Therefore, it is necessary to conduct a further study by considering routes with various features. Secondly, participants did not navigate through the route with their regular aid. The lack of empirical comparison with the regular aid may have led to influencing their critique of the proposed system (*e.g.*, some of them may have completed the route with their cane and rated their regular aid better in Likert questions). In addition, there were limitations regarding the recruitment of participants. First, we were not able to recruit younger participants from the target population, and the number of guide dog users recruited in the main study was not sufficient. Also, while several participants participated in user studies for the first time in our institution, others have not. For whom have participated in the user study in our institution in the past, there may have had a positive bias to our study.

Limitation of Form Factor

In this study, we used a wheeled robot for its advantages, but the form of the wheeled robot may induce several limitations on real-world usage. PathFinder may not be able to navigate through uneven terrains or outdoor environments due to the small size of the wheels. In such cases, we should consider using larger wheels and stronger motors. Also, users would have to carry the suitcase when navigating through the stairs, which could be physically demanding. We believe the gradual improvement of weights and capability of each component used would ease these issues.

In addition, there are also other problems that have to be solved. Firstly, in our implementation, the smartphone and the computer of the system were connected through Bluetooth. Interference by the surrounding usage of Bluetooth may cause the connection to be disturbed. In actual deployment, a robust way to connect the smartphone and the computer is required. Secondly, as the system keeps recording the surrounding environment, the privacy of pedestrians may be compromised. Although it has been shown that sighted people accept the usage of RGB images as long as they are used for assisting blind people to some extent [60], this problem should be carefully considered when actually deploying the system in the real

world. Finally, PathFinder will not be able to navigate in a congested space. This is a well-known issue that is still being studied in robotics [206].

5.9 Conclusion

This paper presents PathFinder, a map-less navigation system that navigates blind people to their destinations in unfamiliar buildings. Through this research, we identified that sign and intersection information is necessary for map-less robots to assist blind people in deciding their way in scenarios where they receive route descriptions from sighted passersby. We also showed that even in an unfamiliar building, blind participants were able to decide and navigate their way to the destination by using PathFinder, which was designed based on the requirement investigation. Future work will involve extending the area of navigation, such as to a multi-floor environment, which may lead to additional information needs (*e.g.*, information on elevators and stairs).

Chapter 6

WanderGuide: Indoor Map-less Robotic Guide for Exploration by Blind People

FIGURE 6.1: **Five Core Functionalities of WanderGuide.** The system assists users in recreational exploration by explaining the surrounding environment through images obtained from the robot's camera. Users can adjust the level of detail and ask questions about their surroundings. Additionally, the system can guide users to locations they have visited before.

6.1 Introduction

In this chapter, we tackle the *exploration* task, which is a task that allows one to gain familiarity with novel environments that they do not know (Section 2.1.3). To realize the concept of map-less systems that truly allow blind people to visit various environments, assisting this task is essential. Exploration tasks are more complex than navigation tasks, as they require two processes to be performed simultaneously: navigating and learning the surrounding environment. Many existing works, including the one presented in this dissertation, have focused solely on the navigation task [9, 10], while others have focused solely on the exploration task, but without automated navigation or by using prebuilt maps and infrastructures [13]. Several systems that do not require maps, as well as remote sighted assistance (RSA) [119, 117, 118], have been developed to guide blind people in various locations [24, 22, 20, 95]. However, these systems primarily focus on navigation to target destinations, not exploration, thus providing only navigation-related information to users (*e.g.*, intersections [22, 24] and signs [24]). These systems are also not independent from human assistance. The recent development of computer vision and natural language processing has led to models that can describe various environmental features obtained

from visual input in highly natural language [207, 122], which, combined with existing map-less navigation technology (*e.g.*, PathFinder [24]) has made us consider the realization of this task and motivated us to develop a map-less exploration system.

TABLE 6.1: **Comparison to Previous Work.** Our work explores a unique characteristic that has not been investigated in the past. In the Purpose row, "Navigation" refers to systems primarily designed to guide users to their intended destinations, while "Perception" refers to those focused on understanding the surrounding environment. "Multi-purpose" refers to systems capable of performing various tasks, and "Exploration" refers to those designed for navigating and enjoying facilities, characterized by constantly discovering and changing goals (*e.g.*, window-shopping [36, 13]).

System	Map-less	Purpose	Independent from Human Assistance	Device
NavCog [9]	✗	Navigation	✓	Smartphone
CaBot [11]	✗	Navigation	✓	**Robot**
Tactile Compass [52]	✗	Navigation	✓	Handheld Device
ChitChatGuide [13]	✗	**Exploration**	✓	Smartphone
Kayukawa *et al.* [18]	✗	**Exploration**	✗	**Robot**
Corridor-Walker [22]	✓	Navigation	✓	Smartphone
Snap&Nav [23]	✓	Navigation	✗	Smartphone
PathFinder [24]	✓	Navigation	✗	**Robot**
WorldScribe [125]	✓	Perception	✓	Smartphone
GPT-4o Demo [122]	✓	Perception	✓	Smartphone
MLLM Powered Applications (*e.g.*, Seeing AI [120])	✓	Perception	✓	Smartphone
RSA [118, 117]	✓	Multi-Purpose	✗	Smartphone
WanderGuide	✓	**Exploration**	✓	**Robot**

To bridge the gaps and address the shortcomings of existing systems, we developed a system with the following characteristics as shown in Table 6.1: 1. Our system does not rely on prebuilt maps or preinstalled infrastructure. 2. Our system focuses on exploration. 3. Our system does not require supplementary assistance from humans. 4. Our system can automatically guide users physically during exploration. None of the prior systems possesses the combination of all these characteristics. Given that the design space for a map-less exploration guide robot remains underexplored, this work aims to investigate and establish the key components of such a system. Following the previous chapter, we used the AI suitcase [17] because of its advantage of being able to autonomously navigate blind users and seamlessly blend into the environment. We designed our system to have two features: an automated map-less navigation feature and a surrounding description feature. The automated map-less navigation technology was realized by a waypoint detection algorithm, which predicts navigable points within the environment. Additionally, we equip the robot system with the ability to convey real-time information about the surrounding environment to users using natural language, accomplished through a multimodal large language model (MLLM [122]). Together, these functions allow blind users to learn about the surrounding environment through MLLM-based description while autonomously navigating with the robot's waypoint detection algorithm, realizing an experience similar to navigating with a sighted assistant. Each of these functions corresponds to PathFinder's intersection detection and sign recognition designed for the navigation task, but is specialized for the exploration task; for example, waypoint detection can be regarded as an extension of intersection detection to open spaces to enable movement in a wider variety of locations, and the MLLM-based surrounding description can be regarded as a redesign of sign recognition to convey information not limited to specific objects but for exploration purposes.

The abovementioned system raises two questions regarding system design:

- How should the MLLM describe the surrounding environment to enable effective exploration?
- What kind of additional functionality should the robot have for a better exploration experience?

Using our prototype system, we employed an iterative process with the direct involvement of target users to answer the above question and develop our system. In the formative study, the participants were asked to follow the robot, which was controlled in a Wizard-of-Oz fashion [208], along predetermined routes while listening to the environment descriptions. The study revealed three groups of user preferences in the system's descriptions with respect to varying levels of details in the descriptive information received. It also revealed requirements in certain functionalities, such as revisiting locations where the system had mentioned, specifying directions to proceed, and obtaining in-depth information through question-and-answering (Q&A) functionality.

In the second stage, taking the lessons learned from the first study, we present *WanderGuide*, a map-less exploration system for blind people (Figure 6.1). Taking into consideration the previously discovered three groups of user preferences, the system offers three modes for describing the surroundings: (1) Detailed description — in-depth information with high granularity, (2) Balanced-Length description — balanced level of information, and (3) Concise description — minimal but essential details for obtaining quick awareness. We also implemented various new features based on the feedback received from the first study, which includes adopting a high-resolution fisheye camera for better perception of the surrounding environment, allowing users to verbally interact to query about the environment and set explored POIs to be navigation destinations, and allowing users to use directional buttons to control the robot for navigation towards the direction of interest. Our system is also fully integrated with the automatic mapping, localization, map-less navigation, and obstacle avoidance functions of the wheeled mobile robot.

Finally, we conducted a main user study with five blind participants, who were asked to freely explore two floors of the science museum using WanderGuide. The study aimed to investigate the following:

- The overall exploration experience of blind users and what is further needed for future improvement
- Whether users can independently decide where they want to know more about and where they want to go through exploration with the system.

All participants appreciated the experience of wandering freely without a fixed destination, and they expressed their desire to use the system to explore both familiar and unfamiliar areas. Participants also highlighted the need to incorporate recognition of auditory cues from the environment. Additionally, differences in how they interacted with the system were observed: one frequently used buttons to guide the robot towards their areas of interest, one passively followed the robot, and others often asked questions. We also identified a limitation in the system's MLLM when conveying detailed information about the surroundings, such as identifying specific names of objects, which suggests the need for further development in how we input information into the MLLM for exploration purposes.

To the best of our knowledge, our work is the first to investigate the design space of a map-less system for blind people to explore independently. To this end, we made the following contributions.

- We formulated the requirements for the system through a formative study, such as the ability to adjust the level of description based on user preferences and to guide users to previously visited locations of interest, thereby enhancing the exploration experience.
- We developed a full stack map-less exploration system that consists of a waypoint detection algorithm and an MLLM-based perception interaction system on top of an existing navigation guide robot. Additionally, we integrated several functionalities based on the formative study that facilitates the exploration experience.
- We confirmed key design requirements, such as varying the level of descriptions based on user preferences through a usability study. We also gained further insights into users' interaction preferences and into design implications for improving the system, including better recognition of audio cues.

The codes of the system are publicly available in the following link: https://github.com/chestnutforestlabo/WanderGuide.

6.2 Related Work

Besides the exploration activity for blind users described in Section 2.1.3, this section presents related assistance systems, autonomy and control methods for assistive technologies, and scene description approaches for blind users. These build on Sections 2.2.1, 2.4.3, and 2.5, respectively.

6.2.1 Assistance Systems for Blind People To Explore

Robotic guide systems have the advantage of addressing the mobility challenges of blind people with their automatic guidance capability. CaBot [11], the first guidance robot that adopted the form of a suitcase, guides users to specified destinations while referring to prebuilt maps or using an object detector to convey surrounding information. Among them, some are specialized in exploration [18, 209, 210]. A robot system by Kayukawa *et al.* [18] allows users to explore by interactively setting destinations on a smartphone and by calling a museum guide to explain the surroundings. However, both systems heavily rely on prebuilt maps and operate in limited locations where the destinations are readily available. Ultimately, our goal is to develop a system that does not require prebuilt maps and enables blind people to explore independently, *i.e.*, without relying on staff assistance within the facility.

Navigation systems for blind people that do not rely on prebuilt maps and infrastructure, *i.e.*, *map-less navigation systems*, have also been proposed in prior research. Besides real-time perception outcomes, these systems primarily depend on externally sourced route information, such as prior route knowledge from blind users [95, 22], routes described by nearby pedestrians [15, 175, 24], and analyzed images of floor maps captured in buildings [23]. For example, PathFinder [24] is a map-less navigation robot system designed to guide blind users to their destinations based on predefined routes. The system autonomously navigates users by utilizing an intersection detection algorithm [83] and a sign recognition algorithm [24]. These algorithms enable users to determine the correct direction to proceed at key decision points. The system's evaluation found that it is necessary to include functionality that takes users back to their starting location after reaching their destination when navigating unfamiliar buildings. However, these map-less navigation systems are

tasked with reaching a specific destination and are not suitable for exploration, as they only focus on providing information related to reaching the destination (*e.g.*, intersections and signs [24]). In an exploration scenario, any information about the environment may prove valuable, such as layout information [35]. In our study, we aim to explore the underexplored design space of map-less guide robots for exploration purposes, such as how the system should describe surroundings and what are the task-specific functionality requirements.

6.2.2 Autonomy and Control Methods of Assistant Systems

Researchers have emphasized autonomy, *i.e.*, the ability for users to select destinations and routes according to their interests, as an important factor for exploratory activities [13, 60]. To this end, researchers have investigated various control methods based on user inputs [15, 102]. For example, systems with prebuilt maps adopted selecting destinations from a premade list of stores [9, 60], or via conversation [9, 13]. In the case of map-less systems, researchers have adopted feedback-based closed-loop processes to leverage both human inputs and system control. Examples include users specifying proceeding directions at intersections while the robot provides automated guidance to the next intersections [24, 35, 15, 95, 96]. Zhang *et al.* [102] reported that the preferred level of control by blind people may vary depending on context. Therefore, we examine the level of control between users and robots under the novel exploration context.

6.2.3 Scene Description for Blind People

Knowledge of surrounding information is crucial for blind people to explore [13]. Tools for blind people to understand their surrounding environment have been commercially deployed and the topic remains an ongoing area of research. While researchers have proposed tools using visual captioning [110] and question-answering models [111] to help blind users understand scenes, these often fail to provide accurate descriptions at diverse locations [211]. Alternatively, RSA applications (*e.g.*, Aira [117] and BeMyEyes [118]) have long been a practical aid for providing blind people with surrounding scenes. However, RSA systems are not suitable for our task, as the service quality depends heavily on the sighted assistance provided [119]. RSA services may also not be sustainable for extended use until users feel fully satisfied with their exploration experience. With the emergence of LLMs and MLLMs, scene describing systems (*e.g.*, Seeing AI [120], BeMyAI [121] and GPT4o-demo [122]) have been developed, which enable blind people to understand scenes in diverse scenarios [123, 124]. ChitChatGuide [13] employs LLMs to interpret predetermined maps and deliver exploration-related information during navigation to a specified destination. However, unlike our system, it relies on prebuilt maps and lacks the capability to provide real-time information. MLLM-based systems, such as WorldScribe [125], offer real-time information by analyzing captured images. WorldScribe [125] also adapts the level of description based on user context, such as the speed at which the device is moved. Our WanderGuide similarly provides three levels of descriptions but the selection is adjusted based on individual user preferences rather than situational context. On the system level, the core distinction is that WanderGuide combines MLLM with a navigation robot, allowing users to concentrate fully on the descriptions of novel environments and navigate to interested places. This combination makes WanderGuide particularly well-suited for *exploration while navigating*.

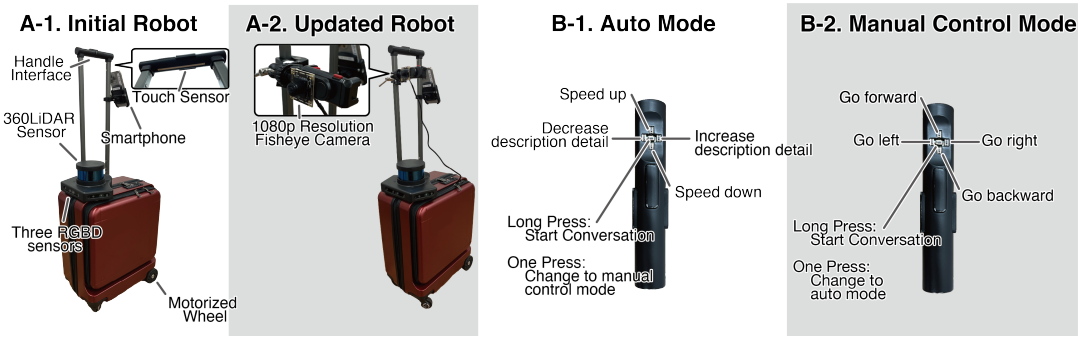


FIGURE 6.2: **Robot and Handle Interface Used in the Study.** Panel A-1 shows the robot used in the formative study, while Panel A-2 presents the robot used in the main study. Panels B-1 and B-2 illustrate the mapping of the handle interface buttons' functions, depending on the selected navigation mode.

6.3 System Design

Our goal is to finalize a system that assists blind people in exploring an indoor environment independently. In this section, we describe the key design elements of the system.

6.3.1 Device

Consistent with the previous section, we chose to use a navigation robot because of their autonomous navigation and obstacle avoidance capabilities. This allows users to concentrate on learning the environment [16, 102, 35]. Our assumption is that the devices should ensure the users' safety during navigation and allow users to focus on exploration. As a result, the findings in our study can be extended to any similar devices other than wheeled robots.

6.3.2 Navigation

WanderGuide adopts a waypoint detection algorithm, which enables the robot to predict navigable points. We designed this algorithm so that the robot could operate in public spaces for exploration, which often include open spaces. It is important to note that PathFinder's intersection detection coupled with collaborative interaction is a design adopted for the navigation task. To assist the exploration task, the system autonomously navigates by predicting navigable places, as blind users do not know where they want to go at the initial stage of exploration. Once the place they want to go is determined as they explore the building with the scene description function, we expect the user to specify that point, and the robot is designed to take them there.

6.3.3 Describing Scenes

Previous navigation systems relied on hardcoded information [9, 13] or simple image captioning models [110] to provide scene descriptions. They only convey information related to navigating to destinations. In exploratory tasks, any information and details could be relevant. Therefore, we decided to use MLLM, a foundational model capable of recognizing a variety of objects and describing them in natural language. We injected MLLM into the system to periodically provide comprehensive information about the surroundings to inform blind users during exploration.

In this paper, we investigate the appropriate presentation format, such as content types and lengths, and the quality of the responses from MLLMs through our user studies.

6.3.4 Interaction

The ability for users to select destinations and routes according to their interests, often referred to as autonomy, is particularly important for exploration [13, 60]. In our system, to what extent users prefer to take control over the robot (*i.e.*, interaction) remains unknown. Based on the scene descriptions given by the system [13], some blind users may fully embrace letting the robot guide them automatically, while others may prefer to decide which way to go on their own. Additionally, this preference may also be influenced by the robot's descriptions of the scenes. Given that the extent of user preference for autonomy remains unclear, we first conducted the formative study (Section 6.4) to explore the requirement of autonomy based on interaction needs. Then, we conducted a full study (Section 6.6) to evaluate the users' opinions on autonomy in our improved system, which integrated the feedback from the formative study.

6.4 Formative Study

We first conducted a formative study to investigate the requirements of the system, such as how the system should explain its surroundings, what further function is required, and what potential interactions may happen between the robot and the user. To conduct the study, we recruited ten participants through our existing email list. Interestingly, our recruitment emails were shared among blind people, eventually reaching people not on our emailing list. In the recruitment email, we specified that those who are unfamiliar with the experimental location, *i.e.*, even if they have had previous visiting experience, they do not have a clear understanding of the building or know what is there, would be eligible to participate. Table 6.2 shows the demographics of the participants. All studies in this paper have been approved by our institution's review board. Informed consent was read out to all participants in this paper and obtained from them. The study took approximately two hours, and the participants were compensated $20 per hour and reimbursed for their transit costs. Only one participant was present for each session.

6.4.1 Prototype System

Apparatus

Consistent with the previous section, we adopted an AI suitcase, which is a wheeled robot [17] as a device. In this study, we used the model called the ACE model, which is visualized in Figure 6.2 A-1. The robot has a handle embedded with five buttons, a touch sensor beneath the handle, a 360° Velodyne VLP-16 LiDAR sensor [212] sensor, three RGBD cameras with resolutions of 640×360, one RealSense D455 camera [213] mounted at the front, two RealSense D435 cameras [214] on the left and right, and a pair of motorized wheels in differential drive configuration. Inside the suitcase, it has Ruby R8 powered by an AMD Ryzen R7-4800U CPU [215], and a Jetson Mate featuring multiple Jetson Xavier NX GPUs [216]. The RGBD cameras were attached 0.51 meters above the ground. The touch sensor detects whether or not the user is holding the handle and moves only when it is being held by the user. The cameras

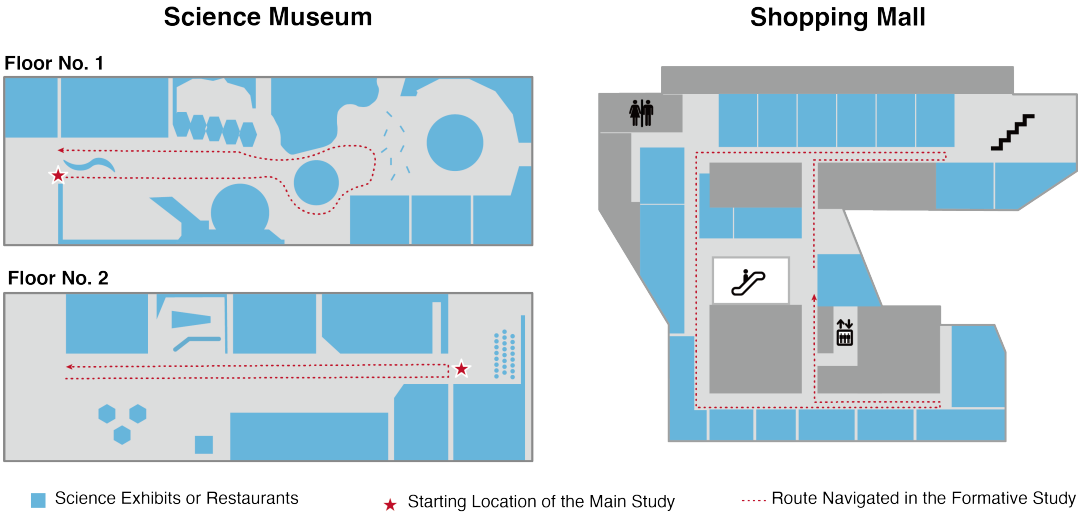


FIGURE 6.3: **Floor Maps of the Study Location** The left panel shows the two floors of the science museum, the fifth floor of Miraikan, which feature exhibits on various topics, such as environmental issues and space exploration. On the right panel is a floor plan of a shopping mall, the fourth floor of Toranomon Hills Station Tower, which includes a variety of restaurants offering different cuisines, including French, Japanese, Chinese, and cafes.

combined have a horizontal field of view of approximately 180°. The weight of the robot is approximately 15kg. We set the default speed of the robot to 0.5 meters per second to maintain a balance between a comfortable walking speed and a speed that allows sufficient time to absorb the scene description audio. A smartphone is attached to the suitcase to provide audio feedback through a neck speaker worn by users, connected via Bluetooth. While the appearance differs from the GT2 model described in Section 5.4.1, the overall functions remain consistent. One notable difference is that there are three RealSense cameras on top of the suitcase, which capture the left, right, and front of the suitcase, whereas the previous model captured only the front. This design enables the robot to visually perceive the environment more extensively.

Scene Description

To convey the surrounding information to the participants, we used GPT-4o [122], a popular MLLM model. We input the images from the three RGBD cameras into the MLLM model and asked the model to generate descriptions of the surrounding environment. The robot was designed to describe surrounding information 5-10 seconds after the end of the previous description every time. We engineered the prompts to ask the MLLM model to first provide a general overview of the scene, followed by specific details on the left, front, and right. We asked the descriptions to include as many objects as possible and incorporate layout information, such as navigable directions and the presence of walls [35]. The processing time and cost to generate a description was 6.087 seconds and $0.00740 on average.

6.4.2 Experimental Location

To ensure the diversity of the findings we would obtain from this study, we conducted the study in two different locations. We chose to conduct our studies in a

TABLE 6.2: **Participant Demographics in the Formative Study.** The table reports their gender, age, navigation aid, which they frequently use, frequency of exploration done either independently or with sighted people per year, their experimental location, number of previous visits to the experimental location, and analyzed preference.

	Gender	Age	Aid	Age of Onset	Frequency of Exploration per Year	Experiment Location	Number of Previous Visits	Preference Analysis
P01	F	64	Cane	44	48	Science Museum	1	Exploration-Inclined
P02	M	53	Cane	13	36	Science Museum	0	Destination-Oriented
P03	M	74	Cane	0	1	Science Museum	0	Destination-Oriented
P04	F	54	Cane	0	12	Science Museum	0	Exploration-Inclined
P05	M	56	Cane	52	2	Science Museum	0	Intermediate
P06	M	32	Cane	0	12	Shopping Mall	0	Intermediate
P07	F	55	Cane	52	0	Shopping Mall	1	Exploration-Inclined
P08	M	63	Cane	22	12	Shopping Mall	0	Intermediate
P09	F	78	Guide dog	22	12	Shopping Mall	0	Destination-Oriented
P10	F	49	Cane	3	1	Shopping Mall	0	Exploration-Inclined

science museum and a shopping mall, as these are locations where people typically engage in exploration, and they have been utilized in previous research [210, 209, 13]. A museum is generally a place for learning about exhibits, while a shopping mall often requires exploration both before and during visits to stores. Specifically, we used the fifth floor of Miraikan[1] for the science museum and the fourth floor of Toranomon Hills Station Tower[2] for the shopping mall. The floor map of the science museum is illustrated in the left panel of Figure 6.3, which contains two floors, both primarily featuring science exhibits. For the studies, the order of the two floors was counterbalanced. The study in the museum was conducted after business hours, during which customers were absent, but staff were present for their duties. The floor map of the shopping mall is illustrated in the right panel of Figure 6.3, a floor that contains several restaurants from various countries. The study in the shopping mall was conducted during regular business hours. As shown in Table 6.2, the study with P01–P05 took place in the science museum, and the study with P06–P10 took place in the shopping mall.

6.4.3 Procedure

For each participant, we first conducted a pre-study interview to learn about their experience in exploring buildings, followed by an explanation that the study aimed to gather their opinions on a guide system designed to assist with exploration. Then, participants were given a task to navigate the predetermined route (red arrow of Figure 6.3) guided by the robot. To focus on the research question and ensure that the participants experienced the same level of autonomy, we adopted teleoperation using a Wizard-of-Oz-based approach [208], in which an experimenter controlled the robot to operate in full-automatic mode by navigating along the route and stopping when there were nearby pedestrians. During exploration, the robot periodically generated descriptions of the scenes. We show an example of the generated description in Figure 6.4. After the exploration, we asked the participants if there were any additional things they wanted to do to partially simulate the potential interaction, such as going to additional places or going around the floor again for more exploration. Finally, we conducted a post-interview session to gather their feedback on the system.

[1] https://www.miraikan.jst.go.jp/en/
[2] https://www.toranomonhills.com/

6.4.4 Result

Interests to Exploration

All participants stated that totally independent exploration is challenging, but they expressed a desire for exploration if a guide system can help them do so. For example: **C6.1**:*"I don't really explore much. I go out with a specific purpose in mind [...] The reason is that it's just too bothersome. But I do think it would be fun if I did [...] I'm more of an old-timer, so exploration never really caught my interest. It's not that I didn't care at all, but perhaps I've been living this way (not to explore).* (P02)

Positive Feedback and Appreciated Information

Seven participants (P01, P04–P08, and P10) expressed their enjoyment while navigating with the robot, particularly with the provided surrounding descriptions, as described in the following comment: **C6.2**:*"My first impression was that it was a lot of fun. The reason is, as you just mentioned, unlike the person I usually walk with, the system provided detailed explanations about things like the color of the walls and the signs we saw and even described how the chef was preparing the food. Normally, you might get some of this information from others, but it's rare to get such thorough details. I found myself thinking, "Oh, I see, that's how it looks to sighted people," and I felt there was a lot of new information. In that sense, I really enjoyed it."* (P07)

Participants appreciated a variety of real-time details about their surroundings, notable examples include patterns on the walls, lighting conditions, subjective descriptors such as "beautiful," the presence and actions of nearby people, the existence of signboards, the layout of the environment, and the visibility of a chef in an open kitchen. Additionally, P10, who requested to walk around the floor again, noted that receiving different descriptions of the same location was beneficial, as it gave them a sense of presence: **C6.3**:*"The system mentioned those things, as well as details about the plants and wall decorations. It's like, you talked about so many different things that it feels like I was actually looking around myself. Honestly, most of the time, I get so occupied with just reaching my destination that I don't notice things around me. [...] The system also mentioned things in the second round of explanations that weren't covered in the first round, which was nice. It conveyed a sense of the ongoing atmosphere and gave a good understanding of the situation at the time."* (P10)

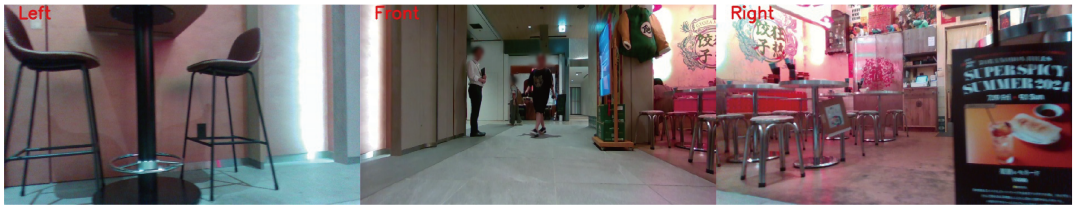
Information Needs

Participants hoped for further polishing of the delivered information about the scenes. Six participants (P01–P03, P06, and P09–P10) felt the information conveyed about the surroundings was too abstract, indicating the need for more specific information: **C6.4**:*"The system talked about there are just exhibits, or there's information on panels, but I think it would be nice if the system talked about specific titles. There are places where the system talked about them, but there are also places where it did not, so I found myself wondering about that."* (P01) In particular, three participants (P02, P03, and P09) commented that the descriptions neither helped them learn the environment nor make decisions such as determining which shops or exhibits to enjoy: **C6.5**:*"I expected it to at least tell me the name of the store, but it was disappointing to find out that it didn't do that at all. I really wish there was a system that could provide pinpointed information about what I want to know. Especially in an unfamiliar restaurant area, for example, if I come alone and use the device to enter the premises, it starts running, and then when I think, "Oh, should*

A. Example at Science Museum

"This is a futuristic exhibition hall that has vibrant displays. To your left, there is a uniquely shaped wooden table and archway. Ahead, you can see a curved blue sofa and a white sign that reads "Entrance." On your right, large colorful panels line the wall, displaying information about the future and health."

B. Example at Shopping Mall



"This is a bright, modern corner of a commercial facility. On the left side, there are tall-backed chairs made of black metal lined up, and beyond them, round tables are arranged. Ahead, a man in a suit is standing, and in the background, there's an electronic menu board, suggesting the presence of a restaurant. To the right, there's an eatery enclosed by warm-colored walls in shades of red and orange, with many metallic chairs and tables, and menu boards are set up."

FIGURE 6.4: **MLLM-Based Description Examples.** Panel A shows an example of a description generated at the science museum, and Panel B shows the one generated at a shopping mall.

I have Japanese food today, or maybe tonkatsu?", without such information, I end up just walking around aimlessly." (P09)

Participants also described specifics about what types of information would be beneficial to include, such as the position of objects given in meters and clock directions, the availability of seats, people on collision paths, identities of surrounding individuals (*e.g.*staff), and specific names of objects. In science museums, participants also wanted to know whether exhibits are touchable. In shopping malls, participants also wanted to learn the store menus and whether there is a spacious area for a guide dog to rest while the user is eating. However, three other participants (P02, P03, and P09) found certain information, such as details about lighting, surrounding people, and wall design, unnecessary.

6.4.5 Design Considerations

The results of the study affirmed that there are certain appreciations and room for improvement for the exploration robot for blind people. Based on the above results, we derived several requirements for the system, as listed below.

Vary Detail of Descriptions Based on Preferences and Contexts

We observed three types of preferences: one that enjoyed all the descriptions provided by the system (*Exploration-Inclined*), another that enjoyed the descriptions but preferred to limit certain information (*Intermediate*), and a third group that only wanted information useful for determining where to go (*Destination-Oriented*). In Table 6.2, we show the description preference of each participant. To classify the preferences, we first classified three participants who did not enjoy the description

of the system as *Destination-Oriented*. Then, based on the discussion between the authors, we classified the rest as *Intermediate* or *Exploration-Inclined*. Furthermore, the type of information needed varied slightly depending on the experimental location. For instance, participants sought seating information for guide dogs in shopping areas, whereas in the science museum, they were more interested in whether the exhibits were touchable. Given these three types of preferences and context-dependent information needs, we modified the system so that it could adjust the amount and types of information conveyed to each participant.

Add Question and Answer Functionality

There was a clear need for question-and-answer (Q&A) interaction, as seven participants (P02–P05 and P08–P10) noted that they would like the option to ask more detailed questions through conversation. Participants expressed interest in this functionality when they were curious about the system's descriptions. This would allow them to ask more detailed questions about the objects of interest.

Add "Take-Me-There" Functionality

Four participants (P02, P04, P06, and P10) mentioned that they would like to revisit locations they found interesting after walking around the floor. Example situations include deciding to visit a shop, engaging with touchable exhibits, or returning to chairs discovered during the exploration. In unfamiliar locations, where users may lose their sense of direction, participants also expressed the need for a feature that guides them back to their initial location [24].

Vary Speed and Be Able to Stop the Robot

While the majority found the default speed appropriate for listening and understanding the described information, there were requests for customizable speed settings. Eight participants stated that the robot's speed was appropriate for exploring. Two participants (P04 and P06) expressed a preference for a faster speed. P01 additionally wanted to stop when the robot read out the descriptions of interest. In conclusion, users who are *Destination-Oriented* or have already determined the destination through exploration may want to increase the speed, while users who prefer to take time exploring might wish to slow down or stop the robot entirely.

Add Direction Specifying Functionality

Participants expressed a desire for more active engagement by specifying the movement direction themselves. Four participants (P02–P05) mentioned that they wanted more active control over the movement direction based on their interests. Additionally, we extrapolated that instead of simply following the robot, some users may prefer to interactively choose the direction based on the audio description of the surroundings. This could lead to greater autonomy because it would enrich the exploratory experience by aligning the robot's movement with the users' real-time curiosity and needs, creating a more personalized and engaging exploration experience.

6.5 WanderGuide Implementation

In this section, we provide the implementation of WanderGuide informed by the formative study. Below is a summary of updates made from the implementation of the formative study.

- Attachment of a new fisheye camera for a better view (Section 6.5.1)
- Implementation of a waypoint detection algorithm for realizing autonomous map-less navigation (Section 6.5.2)
- Implementation of three levels of description based on user preferences (Section 6.5.3)
- Implementation of "Take-Me-There" Functionality (Section 6.5.4)
- Implementation of two navigation modes automatic navigation mode and manual control mode (Section 6.5.5)
- Implementation of an interface to adjust speed, level of description, and navigation mode (Section 6.5.5)
- Implementation of Q&A Functionality (Section 6.5.5)

6.5.1 Hardware Update

One of the notable user feedbacks was the need for more detailed information, such as the names of POIs. However, the cameras in the prototype system were mounted at only 0.51 meters above the ground, had low resolution, and had a limited vertical field of view, making it difficult for the MLLM model to consistently capture details. Thus, as illustrated in Figure 6.2–A-2, we attached a fisheye camera with 1080p resolution and a wide field of view to the higher part of the robot.

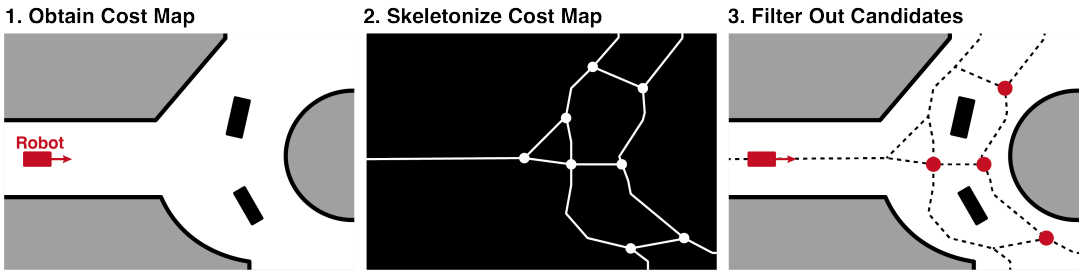


FIGURE 6.5: **Waypoint Detection Algorithm.** Step 1 shows the generated cost map, while Step 2 depicts the skeletonization process of the cost map along with the detection of intersection points. Finally, Step 3 highlights the selected intersection points, which are identified as waypoint candidates.

6.5.2 Waypoint Detection and Navigation

In order to produce destinations to navigate to for the users, a waypoint detection algorithm (Figure 6.5) is necessary to determine navigable points for the robots. As no prebuilt maps were available, we first constructed a cost map, a two-dimensional occupancy grid that assigns costs based on obstacles, and updated it in real-time. We utilized the existing open-source Cartographer package [217], which is a real-time Simultaneous Localization and Mapping (SLAM) algorithm, to generate the cost map.

Next, the cost map was skeletonized, and intersection points on the skeleton were identified based on a kernel-based corner detection algorithm [218]. The intersection points, which are typically far from obstacles, were next used to select potential waypoints. To maintain sparsity among waypoints, we applied the DBSCAN clustering algorithm [219] over the intersection points, and selected the centers of the clusters as potential waypoints. In addition, coordinates three meters in front, behind, and to the sides of the robot were also considered potential destinations to address the case where no intersection points were detected through the algorithm. As selecting a waypoint too far may be challenging for the robot to find a suitable path and a waypoint too close would lead to frequent destination changes, we filtered out candidates further than 50 meters and closer than one meter to the robot. After filtering, the final list of candidate waypoints was set.

During navigation, the robot automatically selected its goal from the candidate waypoints. By default, priority was given to waypoints lying in the same forward direction as the robot's initial orientation, where it was placed and activated. If no forward waypoints are available, the robot selects the waypoint with the smallest absolute angle relative to its current orientation. Once a waypoint was chosen from the candidate list, the robot navigated to it by using the onboard open-source navigation algorithm [11]. Once the waypoint was reached, the next waypoint was chosen automatically using the same process. It is important to note that prioritizing the robot's initial orientation was based on the assumption that users can adjust the general direction to proceed, such as starting from the entrance into the building.

6.5.3 Scene Description Generation

The basic algorithm for scene description generation remains unchanged, but the description was conveyed only when the robot was moving. Also, the MLLM took the overall view image from the fisheye view camera in addition to the three RGB images from the RGBD cameras. According to the results of the formative study, we added three levels of detail in the scene description.

- *Detailed Description* This mode provided rich, immersive descriptions for blind users who wanted to explore their surroundings in detail. The MLLM generated 3-4 sentences (120-240 characters), covering lighting, signs, layout, nearby people, and subjective descriptors like "beautiful" or "modern". The description began with an overview, followed by details of the left, front, and right.

- *Balanced-Length Description* This mode offered clear descriptions for users who preferred concise but informative content. The MLLM generated 2-3 sentences (60-120 characters), focusing on relevant details like signs and layout, while omitting lighting conditions or subjective descriptors. Descriptions covered the left, front, and right, without the overview.

- *Concise Description* This mode provided brief, essential information for users who wanted quick guidance. The MLLM generated 1-2 sentences (less than 60 characters), focusing only on key details needed to navigate, excluding unnecessary information. Descriptions covered the left, front, and right, without the overview.

For MLLM, these three levels were controlled via prompts. All prompts shared the following instructions in common: to convey environmental information that assists blind people to explore, to refer to specific details such as genres or the names of objects, to encourage reading any text that helps users explore, to describe spaces for

guide dogs to sit in restaurants, to provide information about potential hazards, and to use numbers to indicate the relative positions of surrounding objects. To ensure that the MLLM adhered to the instructions provided in the prompt, we employed a two-stage inference process. First, we instructed MLLM to perform an initial inference, generating a description. Then, it self-supervised this generated description to verify if it met the given instructions. Finally, MLLM produced a revised version of the description to be presented to the user. Although this approach resulted in longer inference times, the outputs produced follow complex prompt instructions. The description is read aloud every 5-10 seconds after the previous description has been read out. The processing time and cost to generate a description was 5.78 seconds and $0.00811 for a Detailed Description, 4.75 seconds and $0.00753 for a Balanced-Length Description, and 4.02 seconds and $0.00734 for a Concise Description on average.

6.5.4 "Take-Me-There" Functionality

Acting on the feedback received from the formative study, we implemented a function that guided users to a destination verbally specified. This feature was typically enabled by the robot's *semantic map* [220, 221, 55]. In our case, we linked the images and generated descriptions, which had been saved as the robot had navigated, to the cost map of the robot. Given a verbal cue from the user (*e.g."I want to go to the blue sofa."*), the system first used our selected MLLM model to extract the name of the target location (*e.g.*, blue sofa). Then, we calculated the embeddings of the target location, all saved captured images, and all saved generated descriptions. We took a dot-product similarity between the extracted target location and the embeddings of images and descriptions to find the closest match. We used pre-trained feature extraction models: a fine-tuned SimCSE [222] model for generating sentence embeddings from text and a pre-trained CLIP [223] model for creating image embeddings. We used models that were trained in the native language where the study was conducted. The coordinate linked to the closest matched image or description would be set as the destination. If the user wanted to go back to the initial location, we used MLLM to detect the user's intent and set the destination to the initial point. We note that a similar functionality, the "Take-Me-Back" functionality, which allows users to return to their initial location, has been implemented in the previous map-less navigation system PathFinder [24]. The "Take-Me-Back" functionality is specifically designed for navigation purposes, as it was motivated by the challenge blind individuals face in returning to their original location after navigating. In contrast, our functionality is tailored for exploration tasks, enabling users to return to any point of interest they identified during their exploration. Ultimately, our functionality encompasses the capabilities of the "Take-Me-Back" feature while extending its application to support exploratory activities.

6.5.5 Navigation Mode and User Interface

On the high level, we implemented button controls and conversation interaction methods for users to interact with the robot.

Button Controls

We utilized the four directional buttons and the central button on the handle of the suitcase-shaped robot to enable users to control the robot's speed, adjust the level of

TABLE 6.3: **Participant Demographics in the Main Study.** The table reports their gender, age, navigation aid, which they frequently use, frequency of exploration done either independently or with sighted people per year, their experimental location, and number of previous visits to the experimental location.

	Gender	Age	Aid	Age of Onset	Frequency of Exploration per Year	Experiment Location	Number of Previous Visits
P11	M	59	Cane	29	0	Science Museum	1
P12	F	59	Cane	43	36	Science Museum	1
P13	M	56	Cane	45	1	Science Museum	0
P14	F	60	Cane	45	48	Science Museum	1
P15	M	59	Cane	24	4	Science Museum	0

descriptions, switch between automatic and manual control modes, and specify the direction of movement. The mapping of the buttons is illustrated in Figure 6.2–B-1 and B-2. The central button was used for mode changes. The functions of the directional buttons would change depending on the robot's modes: *auto mode*, *manual control mode*, and *conversation mode*. In auto mode, the robot navigated by determining the waypoint automatically. The left and right buttons allowed the user to switch between three levels of description, where the default mode is the balanced-length description mode. The forward and backward buttons were used to adjust the robot's speed. Users can adjust the speed from zero to one meter per second, with increments of 0.05 meters per second. In manual mode, users could specify directions on their own. The robot would instruct the user to press the directional buttons to select the direction to proceed. If there was a suitable waypoint in the specified direction, the robot would inform the users via voice feedback. Otherwise, the robot conveyed that there were no navigable points in the specified direction. In conversation mode, triggered by long-pressing the central button, the robot would pause, and all four directional buttons were disabled until the conversation was ended. Users could manually end the conversation by long-pressing the central button again. The details of the conversation mode are described below.

Conversation

The conversation mode allowed users to give commands or ask questions with verbal input via the smartphone attached to the robot (Figure 6.2). When the user inputted their verbal cue, the system used MLLM to classify the user's intent into one of three categories: usage of "Take-Me-There" functionality, usage of Q&A functionality, and direction specification. If the detected intent was direction specification (*e.g.*, *"I want to go to right"*) the robot would navigate to the waypoint in the specified direction accordingly. Finally, users could finish the conversation with an ending phrase such as *"Thank you."*

6.6 Main User Study

This study was conducted to investigate the overall exploration experience of blind users, further design space, and whether users can independently decide where they want to know more about and where they want to go through exploration with the system. Participants were recruited and compensated similarly to those in the formative study. Similar to the formative study, in the recruitment email, we specified

that participants unfamiliar with the experimental location would be eligible to participate. We conducted this study on the same two floors of the science museum. Table 6.3 shows the demographics of the participants. None of the participants from the formative study participated in this study. Similar to the formative study, this study was conducted after business hours.

6.6.1 Task and Procedure

For each participant, we first conducted a pre-study interview similar to the formative study. Then, the participant joined a 30-minute training session to get familiar with the robot system before the main tasks. For the main tasks, they were asked to freely explore the floor for 20 minutes using the system from a fixed starting location, as illustrated in Figure 6.3. The ordering of the floors was counterbalanced to mitigate the order effect. After the main tasks, we conducted a post-study interview to ask several seven-point Likert scale questions (1: Strongly Disagree, 4: Neutral, and 7: Strongly Agree) that measure their self-evaluated exploration performance, Raw Task Load Index (TLX) [224] to measure the task workload, and system usability scale (SUS) [174] to evaluate the usability of the system. Finally, we asked open-ended questions to gather comments on the system. Below, we report the results of the study.

TABLE 6.4: **Activity Breakdown.** The table shows the statistics of duration time and the count of interactions for each mode (Auto, Conversation, and Manual Control). The ratio of the duration time is calculated based on the total duration time of the experiment per participant.

	Auto		Conversation		Manual Control	
	Ratio(%)	Count	Ratio(%)	Count	Ratio(%)	Count
P11	59.77	25	37.52	21	2.70	4
P12	91.66	13	8.16	9	0.18	1
P13	67.88	12	30.56	9	1.56	1
P14	64.86	17	33.94	15	1.20	1
P15	58.53	28	20.03	19	21.44	10

6.6.2 Analysis of Participants Activity During The Task

We report the statistics of each participant's activity during the task by referring to the system's log and the video captured during the tasks. Table 6.4 shows the analysis of their time spent on the three modes as specified in Section 6.5.5. We noticed that the activation quantity and duration of each mode varied significantly among participants. P11, P13, P14, and P15 frequently used the conversation mode. Notably, P11 spent nearly 40% of the total time engaging in conversation with the robot. In contrast, P12 barely used the conversation mode and relied on the auto mode for 90% of the total time. P15 was the only participant who actively used the manual control mode.

6.6.3 Analysis of Requests from Participants During Within The Conversation Mode

In Table 6.5, we further report the statistics of requests from participants within the conversation mode. Note that the total count of conversations in Table 6.5 is bigger

TABLE 6.5: **Requests Breakdown During Conversation Mode.** We defined three types of queries: General Query, Specific Query, and Command Query, and classified each participant's request into one of these categories. Among Command Queries, requests were further classified into going to a specific location (*i.e.*, the command that triggered the "Take-Me-There" functionality), returning to the initial location, going toward a specific sound source, and other requests (*e.g.*, requests that the robot cannot execute, such as going along the wall). The ratios represent the percentage of each category over the total number of requests.

	General Query	Specific Query	Command Query				Total
			Specific Location	Initial Location	Sound Source	Others	
P11	11%	46%	14%	0%	11%	17%	35
P12	30%	10%	50%	10%	0%	0%	10
P13	62%	38%	0%	0%	0%	0%	13
P14	14%	46%	25%	0%	0%	14%	28
P15	8%	25%	46%	4%	0%	17%	24

than the conversation mode counts in Table 6.4, as multiple turns of conversation could happen in one conversation mode interaction. We classify each verbal request into three categories.

General Query Request general information in the surrounding area or in a particular direction.

Specific Query Request detailed information about a specific object in the environment.

Command Query Issue command to guide to destination, triggering "Take-Me-There" functionality, direction specification via conversation, or other commands users ask the robot to do.

Among Command Queries, there were four classifications:

Specific Location Query A command to go to a specific location, triggering the "Take-Me-There" functionality.

Initial Location Query A command to return to the initial location, also triggering the "Take-Me-There" functionality.

Sound Source Query A command to go toward a specific sound source, which cannot be executed by the robot.

Others Other commands that cannot be executed by the robot, such as a request to go along the wall.

Overall, we discovered that although our system constantly provided environmental descriptions in auto mode, users still preferred to ask for general information about their surroundings or in a specific direction in conversation mode. For example, P13 predominantly made General Queries (62%). Users also had diverse preferences when using our system. Some users such as P11 (46%), P13 (38%) and P14 (46%) were interested in learning the specifics of POIs, reflecting the takeaways obtained in Section 6.3. Some users such as P11 (43%), P12 (60%), P14 (39%), and P15 (67%) favored using conversation mode to instruct the robot to guide them to their destinations. In particular, by referencing Table 6.4, we can see that P11, P12, and P14 preferred conversation mode over manual control mode to issue commands. This validates the extrapolated idea in Section 6.4.5.

Among the Command Queries, we observed that four participants used the "Take-Me-There" functionality. This indicates that the MLLM-based description triggered their interest in specific locations and motivated them to go to those locations, demonstrating the potential of this system to enable exploration activity for blind people. Interestingly, although we explicitly informed participants during the tutorial that the executable interactions were limited to the "Take-Me-There" functionality and the Q&A functionality, they asked the robot to perform actions that were not executable, such as going toward a sound source they found curious or keeping along the wall. This behavior highlights further potential improvements that need to be made to the system.

TABLE 6.6: **Error Analysis of MLLM.** We classified the errors into five categories and counted the number of them. Note that a single response could contain multiple errors, so the sum of errors does not match the total output of MLLM.

	Wrong Character Recognition	Wrong Object Recognition	Nonexistent Objects and Texts	Misunderstanding User Input	Inaccurate User Input	No Error	Total output
Scene Description	31	6	11	-	-	117	164
Q&A Response	9	6	15	5	1	21	53

6.6.4 Error Analysis of Scene Description and Q&A Responses

In Table 6.6, we report the accuracy of MLLM responses both during auto and conversation modes. We manually analyzed the text output generated by MLLM and compared it with the logs of the images saved. We classified and counted the errors made by MLLM into six categories.

Wrong Character Recognition Misrecognition of text, such as misreading signs.

Wrong Object Recognition Misidentification of objects in the scene.

Nonexistent Objects and Texts Mistakenly recognizing objects or text that are not present. Note that this differs from the previous two categories, where some similar objects or text were actually present.

Misunderstanding User Input Misinterpreting a user's question in conversation mode, such as providing an environmental description when asked to read text from a panel.

Inaccurate User Input Errors made when the user asked about objects or text that were not present.

No Error Accurate responses with no errors.

When multiple errors occur in a single sentence, errors of the same type are grouped together and counted as one. Errors of different types are counted separately. For instance, if there are multiple text recognition errors in a single sentence, they are counted as one text recognition error. If a sentence contains both text recognition errors and object recognition errors, each is counted separately as one text recognition error and one object recognition error. Thus, note that the total number of errors may not match the total number of outputs.

The results showed that 28.6% of the outputs contained some form of error during scene descriptions whereas 60.3% of conversation mode outputs had errors. This

difference is likely because users in conversation mode often asked for more detailed explanations, which led MLLM to attempt more complex responses and, as a result, made more mistakes. This was particularly evident in the *Nonexistent Objects and Texts* category, which accounted for only 0.07% of errors during scene descriptions but significantly higher at 28.3% in conversation mode. This means that MLLM often generated descriptions of objects or text that did not exist in the environment when asked for more detailed information. Character recognition errors were common in both modes, likely due to MLLM's limitation in reading distant text. In a general sense, instead of complete failures, MLLM often partially misread the text or misidentified objects with similar-looking ones (*e.g.*, mistaking a tall table for a reception desk). Nevertheless, over 70% of responses in the auto mode were accurate, demonstrating the overall usefulness of the system.

TABLE 6.7: **Description Level Analysis.** The statistics of the usage of each description level. Usage statistics for each description level are calculated by normalizing the duration of each level with respect to the total duration of the experiment.

	Concise	Balanced-Length	Detailed
P11	0.10%	76.46%	23.43%
P12	15.22%	29.88%	54.91%
P13	0.19%	49.14%	50.66%
P14	0.15%	87.94%	11.91%
P15	0.14%	99.86%	0.00%

6.6.5 Analysis of Usage of Each Description Level

In Table 6.7, we report the statistics of how much time participants spend their time using each description level. The result shows that there were three types of usage during the study. P15 only used Balanced-Length mode, P11 and P14 used Balanced-Length mode most of the time while sometimes using Detailed mode, and P12 and P13 used Detailed mode most of the time. This shows an interesting trend, as this function was initially designed to adapt to each user's preference and therefore, should not have been changed during the task. However, in practice, participants dynamically adjusted it during the task to adapt to the context. As described later in C6.11, this suggests that the design addresses not only preference but also contextual factors.

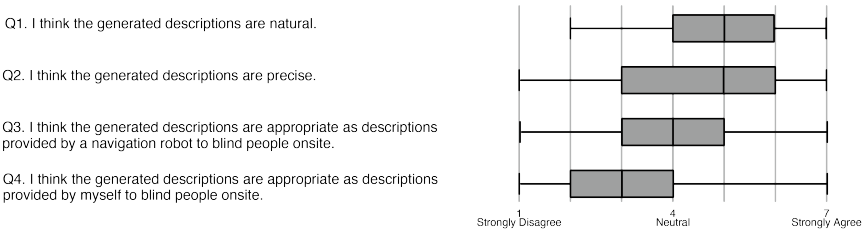


FIGURE 6.6: **Qualitative Evaluation with Human Experts.** The table shows a box plot of the evaluation with human experts in seven-point Likert points. The results indicate that they generally felt the descirption natural and precise but was not yet suitable for onsite explanation.

6.6.6 Scene Description Quality Evaluation

Finally, to analyze the quality of the MLLM-generated scene descriptions from the human expert perspective, we conducted a survey with human museum guides and asked them to evaluate using a seven-point Likert scale. The participants were presented with images captured by the robot, each accompanied by its corresponding generated description, and were asked to evaluate the descriptions in a survey, as shown in Figure 6.6. The survey was conducted in a counterbalanced manner to mitigate potential biases. During the main study, 164 descriptions were generated, and we randomly sampled half (82) of the total descriptions for evaluation. The randomly sampled descriptions contain mixed levels of detail. Each description is evaluated by three to four participants. In total, 56 museum guides participated in the evaluation, with each randomly assessing five descriptions. There were 32 males and 20 females, and four participants did not report their gender. Their average age was 39.6 years, with an average of 5.9 years of experience as a museum guide. On seven-point Likert scale items, the median self-reported familiarity with museums was 5.0, and the familiarity with LLMs was 4.0 (1: very unfamiliar, 4: neutral, and 7: very familiar). Our analysis revealed that the experts generally perceived the generated descriptions as somewhat natural (Q1) and precise in describing an image (Q2) as shown by their median of five. Meanwhile, they found the generated descriptions less suitable as image descriptions for blind people (Q3) and as onsite descriptions provided by experts for blind people (Q4).

TABLE 6.8: **Qualitative Analysis.** Rating to seven-point Likert score questions (1: strongly disagree; 4: neutral; 7: strongly agree).

	P11	P12	P13	P14	P15	Median
Q1. I was able to explore the facility.	4	6	4	6	6	6
Q2. I was able to enjoy the exploration.	4	7	4	6	6	6
Q3. I was able to gain an interest in the things around me.	4	6	6	6	6	6
Q4. The interface of the system was easy to understand.	5	6	6	5	5	5
Q5. I want to explore where I am familiar with this system.	7	7	6	7	6	7
Q6. I want to explore where I am unfamiliar with this system.	7	6	6	7	6	6

TABLE 6.9: **Workload Assessment with RAW TLX.** Lower total scores indicate a lower workload. Each item is scored on a scale from 1 to 10, where 1 represents a lower level, and 10 represents a higher level of Mental Demand, Physical Demand, Temporal Demand, Effort, and Frustration. For Performance, 1 indicates good performance, and 10 indicates poor performance.

	P11	P12	P13	P14	P15	Median
Mental Demand	2	2	5	3	2	2
Physical Demand	2	2	3	2	6	2
Temporal Demand	2	3	1	5	2	2
Performance	7	2	5	7	5	5
Effort	3	2	5	6	6	5
Frustration	8	4	1	3	7	4
Total Score	24	15	20	26	28	

6.6.7 Usability and Workload Evaluation

In Table 6.8, we report the results of seven-point Likert items. For Likert items, a median score of five or higher indicates that participants generally responded positively. The total SUS for P11 to P15 were 72.5, 80, 90, 82.5, and 77.5, respectively, showing acceptable usability of all being above 70 [225]. The total Raw TLX scores for P11 to P15 were 24, 15, 20, 26, and 28, respectively. We show the distribution of Raw TLX scores in Table 6.9. Raw TLX [224], a simplified version of NASA TLX [226], is known to have a high correlation with NASA TLX, and the total NASA-TLX scores for people with special needs typically ranged from 26 to 48 in previous research [227]. Overall, our total Raw TLX scores may suggest that participants did not experience a significant load during the task. We also observed that the median value for mental, physical, and temporal demand was relatively lower, scoring 2. This is likely due to the robot navigating them, allowing participants to explore without being burdened by these demands. Nonetheless, a relatively higher median value was observed for Performance, Effort, and Frustration, indicating that some users experienced a lack of satisfaction with the exploration experience provided by the system.

6.6.8 Qualitative Analysis

Positive Feedback

All participants expressed their appreciation for the experience of wandering around a building to explore without specific destinations in mind with the help of our system: **C6.6**:*" When the camera explains things it recognizes, like how bright the room is or what the floor looks like, or what objects are placed where, I found myself nodding in agreement multiple times, like, "Oh, so this is how it looks." I remember when I first held the suitcase robot, I deeply empathized with guide dog users. I thought, "Oh, so this is what it's like to have a guide dog." However, since I can't take care of a guide dog, I've given up on that option. And now, with this navigation system that explains various situations, it's exactly what I need. It's not just about setting a destination and getting there but feeling the freedom to explore spontaneously. For example, the ability to roam a large shopping mall freely and explore on a whim feels like true freedom to me. Instead of pre-planning every move or relying on a guide, I could simply grab my suitcase and decide to venture out spontaneously."* (P12)

The same participant, P12, who had been to the facility previously, noted that they still had new discoveries with the system: **C6.7**:*" I've been to this museum before, but when the guide explained things to me back then, it was more like a general explanation about the atmosphere and such. But earlier with the system, there was a very detailed explanation that came out of the suitcase. Like, about how bright sunlight comes [...] There were things I didn't know that made me learn new stuff, even though I thought I knew about the facility."* (P12) Also, P12 and P14 noted the feeling of relief not relying on sighted assistance: **C6.8**:*" I don't think there has ever been a system that explains your surroundings while walking. [...] When walking with other people, I often find myself feeling a sense of obligation. I worry that they're putting in extra effort to describe things because I can't see. And then I feel like I have to respond to them since they're trying so hard—which can be exhausting. But with this system, I feel I can go strolling by myself."* (P14)

Participants also noted the functionality to go to an aforementioned destination and Q&A functionality particularly useful: **C6.9**:*"(The "Take-Me-There" functionality is) I think it's wonderful. After all, spatial awareness is difficult, so going back to landmarks is very important. If it is accurate, I think it's great because it can be extremely helpful for*

spatial cognition." (P11) and **C6.10**:*"When engaging in a conversation, not knowing what kind of response you'll get, the feeling of unease and excitement that's both a plus and a minus, I think. But I found it really great that you can still ask questions. So even if the response you get doesn't answer your question, or even if it's just "I don't know," the fact that you can at least ask is important."* (P14)

Adjusting Detail of Description

When we discussed their preference in the level of detail of descriptions, all participants described that it would rather depend on the scenario they are in: **C6.11**:*"It might depend on the location, but I know I can get detailed information in Q&A functionality. So, for familiar places, the Balanced-Length mode might be fine. However, there are parts where I'd want the Detailed Description mode for unfamiliar places. For example, switching between modes could be useful, like having Detailed Description mode first for explanations about the room's brightness and how easy it is to walk around. "* (P12)

Comments to Improve the System

Participants suggested various improvements to the system. One particular suggestion was to incorporate functionality for the robot to understand sounds. As the experiment location was a science museum, various exhibits emitted sounds. P13 noted that they would like to inquire about the sound sources, which were not supported by the system: **C6.12**:*"We are extremely sensitive to sounds, and it becomes a point of interest. At a place like the exhibition hall we're visiting this time, various sounds are coming from all directions. This prompts questions like, "What's happening at that sound over there?" Therefore, it would be advantageous if we could ask specific questions like, "What's that sound coming from the right?" "* (P13)

Also, four participants (P11-P13 and P15) found the descriptions from the system still insufficient to explore, as described in the following comments: **C6.13**:*"The place we did the task this time was quite out of the ordinary. Even if you were walking around with my family, I think they would also have difficulty explaining it. Therefore, I felt it might still be somewhat challenging for machines to handle this kind of thing. However, I did feel it was good that I got a sense of what was there. But when it comes to the actual detailed explanations, it was not there [...]"* (P15)

Specification of Proceeding Direction

While we introduced all functionality to participants within the training session, we observed that only P15 used the functionality to specify which way to proceed via a button or conversation. P15 tended to use the functionality when P15 was interested in a specific object: **C6.14**:*"It seems that when I was told, "There's something on the right," I tried to approach toward it because I wanted to get closer when I used something like that."* (P15)

6.7 Discussion

6.7.1 Experience of Using WanderGuide

WanderGuide provided participants with the experience of exploring unfamiliar indoor environments without a specific destination in their minds, mimicking the spontaneous wandering experience of sighted people (C6.6). Participants expressed

a sense of confidence when using the system, noting that it allowed them to navigate independently without relying on traditional tools like white canes (C6.6). Also, through the experiment, we found that participants were able to find exhibits that they found interesting, which led them to trigger the "Take-Me-There" functionality (Section 6.6.3), showing collaboration between users and the robot. This shows that WanderGuide serves as initial evidence of realizing a map-less exploration system. As described by C6.13 and the ratings of 4 from P11 and P13 to Q1 and Q2 in Table 6.8, there still exists the limitation of being unable to describe specific information. Thus, there is a need for further research on how to appropriately convey surrounding information to blind people. Still, the system's ability to deliver real-time descriptions of objects, walls, and spatial layouts enabled participants to form an imagination (C6.6) of their surroundings, sparking their desire to use the system in familiar and unfamiliar environments (Table 6.8 Q5 and Q6). In short, WanderGuide has the potential to provide users with an experience similar to that of navigating with sighted assistants to explore the environment, but the users can explore independently. We believe this research opens a new frontier to the concept of *map-less exploration* guide system for blind people.

6.7.2 Scene Description by MLLM

Our survey in Section 6.6.6 revealed that descriptions by MLLM were rated high for their naturalness and suitability for general image description but were not for actual descriptions to be provided to blind people by sighted experts. This may be because the style and content of the generated descriptions differ from those typically provided to blind people during live interactions. For example, museum guides often focus on explaining notable objects or visible exhibits, complementing their descriptions with additional knowledge about the exhibits. In contrast, the generated description often lacked concrete explanation about exhibits and shops, such as their names (C6.4, C6.5, and C6.13). This problem may be more prominent because the study was conducted in a science museum, where each exhibit contains detailed information that is not visually apparent but needs to be explained. On the other hand, from participant feedback, participants noted that MLLM-generated descriptions are comprehensive (C6.2, C6.3, C6.7, and C6.8), and provide them with enjoyment (C6.2) and imagination of vision perception (C6.3). They noted that MLLM provided them with information that they usually do not get from sighted assistants, leading to new discoveries (C6.7). The descriptions provided by MLLM additionally allow blind people to tune in without hesitation and the need to rely on sighted people (C6.8). These results indicate that evaluation from sighted experts may be stricter than that from blind people. As the comprehensiveness of the MLLM-based description was appreciated by the participants, as it enabled blind people to imagine the environment by describing features that they typically cannot obtain from sighted people (C 6.2 and C6.7), we emphasize that balancing this advantage with more concrete information is further required to better support exploration.

6.7.3 Personal Preferences

The studies revealed distinct preferences among participants regarding the levels of detail in the descriptions (Section 6.5.3) and interaction modes (Section 6.5.5). From the formative study, participants were divided into three preference groups, highlighting users' diverse information needs regarding exploration and goal-oriented

navigation (Section 6.4.5). Differences in preferences were mainly attributed to personality traits, because participants who were "Destination-Oriented" (Table 6.2), or were mostly concerned with reaching destinations, mentioned they did not enjoy the detailed explanation of the system and preferred short, concise information. For example, one early blinded participant mentioned that exploration did not interest him, as he had barely done it in his daily life (C6.1). On the other hand, some participants enjoyed imagining the scenes conveyed by the system. Congenital users commented that the descriptions felt as if they were actually seeing the surroundings, while acquired users likened it to their recalled experiences when they could still see. Interestingly, those who particularly enjoyed the system and were "Exploration-Inclined" were all female, while the Intermediate group, who enjoyed exploration but wanted more control over the information provided, consisted mainly of male participants. We note that "Destination-Oriented" users expressed dissatisfaction with the system because they felt the scene description capabilities of the MLLM did not meet their expectations for exploration. Therefore, if the system was improved and was able to convey more concrete information, they might express different opinions.

In the main study, further differences regarding how users interacted with the system were observed. Firstly, we observed that participants adjusted the system's levels of description, demonstrating our design aligns with their needs, which were based on three types of preferences identified in the formative study (Section 6.6.5). The variation in the portions used for each mode further underscores the need for configurable descriptions. Also, how they used the conversation mode varied. Three participants frequently asked questions to the system to gather information about their surroundings (Section 6.6.2), while P15 preferred having more manual control over the robot's navigation. Meanwhile, P12 favored the auto mode, where the robot guided them with minimal intervention. These observations highlight the need to consider customizing to various dimensions of personal preference, from description details to user autonomy, for future development.

6.7.4 Design Implications and Future Development Directions

Two key design implications were observed in our studies. First, allowing the users to control the level of detail in the scene descriptions emerged as one of the most important design requirements. The system may benefit from further *personalization* by users verbally describing their personal information needs as in previous research [13]. Second, participants expressed the need for audio-based recognition capabilities, especially in environments where sound is an integral part of the experience, such as museums (C6.12). The ability to answer questions about sounds and potentially guide users to the sounds' sources would enhance their exploration experience.

On the development side, the primary challenge encountered throughout the two studies was the system's inability to provide detailed information that participants required, particularly regarding the identification of POI-related objects, as described in Section 6.7.2. We attempted to address this by upgrading the robot's hardware, *i.e.*, adding a 1080p resolution fisheye camera to a much higher position. Still, participants found the descriptions lacking in detail and conveyed information somewhat vague, as partially shown by the ratings of 4 from P11 and P13 to Q1 Table 6.8. We deduce that this was because the captured images sometimes did not contain useful information, such as the names of certain objects, or because the

MLLM failed to accurately identify the useful information. As a possible improvement, the robot could utilize history images by selecting the image with the best view to generate descriptions. Also, the robot could utilize, other modalities, such as colored point clouds by fusing camera images with the LiDAR sensor to provide three-dimentional sensor details to MLLM [228, 229]. In conclusion, the MLLM module is the bottleneck of our system's technological development. Similar system development efforts in the future should allocate the most resources to tackling this technological challenge. Still, the issue may be gradually solved as MLLM is the current core area actively developed by researchers.

Another significant challenge in development we encountered is the challenge of running map-less navigation algorithms in diverse novel environments, which requires extensive development. Incorporating vision modalities [230], which we did not use in this study, could potentially enhance the robot's navigation capabilities. Achieving this, however, demands human-level object and layout recognition and real-time processing speed, where further research is required.

6.7.5 Limitation and Future Work

We were unable to examine user preferences over the long term, as participants in our study interacted with the system only for a short duration (20-40 minutes in the formative study and 70 minutes in the main study). Only a small portion of the reliance on concise descriptions may be due to the study's design limiting participants' time to explore. The time constraints may have led users to act on the cost-effective information acquisition. However, if the system is used regularly, users may encounter more situations where they prefer to use the concise mode, as indicated by C6.11. Also, their preferences might change as they become more adept at utilizing it as a tool to query information, which the MLLM is particularly proficient at. Thus, future research should investigate the effects of long-term use of the system.

We conducted two studies in two indoor locations. To capture more diverse needs, future studies should also explore the system's performance in more diverse environments. This may reveal various additional information needs. The usage of the wheeled robot, while beneficial in guiding blind users because it is silent [58], remains a constraint when navigating stairs or uneven terrain. This limitation, however, could be alleviated through user collaboration, such as assisting the robot in getting onto elevators or slightly lifting the robot over small steps. Thus, future research should investigate the system devices's capabilities in different environments, as well as how these robots can address physical limitations by interacting with users. Finally, for the main study, we were unable to conduct it in crowd environments with bystanders potentially obstructing the cameras, because the primary study was conducted in the science museum outside of regular operational hours. Handling crowded environments with robots, even when prebuilt maps are used, remains a significant challenge in the field of robotics [231]. Therefore, in future work, we aim to address the usability limitations of our system in such scenarios by integrating novel algorithms designed to manage crowded environments [231].

The MLLM often made mistakes or referred to non-existent objects, with these errors being particularly noticeable in its responses within the Q&A functionality (Section 6.6.4). The most common misrecognitions involved either partially reading the text or confusing objects with similar-looking ones. However, the performance of the MLLM is not the primary focus of our research. To ensure users receive the most accurate information possible, we will continue updating the MLLM used in the system. Also, some of the image inputs provided to the MLLM may have been

affected by motion blur, potentially leading to a degradation in the quality of the generated descriptions. This issue could be addressed by using cameras that are more resistant to motion blur or by implementing algorithms that detect motion blur and select alternative frames for processing.

Recruitment was conducted through our institution's email list, which includes many participants from previous studies. We acknowledge that these participants may have exhibited a positive bias toward our study, as they had expectations regarding the development of the robot system. Furthermore, we obtained valuable insights from five participants, and involving more participants might have provided additional perspectives. Given the difficulty of recruiting many blind participants, we chose to iterate the study with five participants in each study, rather than conducting a single study with a larger group.

6.8 Conclusion

Towards realizing a scalable map-less guide system that assists blind people in exploring, we developed WanderGuide, a robotic guide system designed to provide real-time descriptions of surroundings and to offer conversation functionalities that allow users to specify their destinations or ask questions. The formative study with ten blind participants revealed that there are three types of preferences over the levels of details of the descriptions generated by the system. In a subsequent main study with five blind participants, all of them expressed appreciation for the experience of wandering freely without a fixed destination, as well as a desire to use the system for exploring both familiar and unfamiliar areas. The study further revealed that including audio recognition would be the immediate next step for developing our system. It also revealed that customizing to diverse user preferences is important and that MLLM is the key bottleneck of the technology development of our system. We hope this research contributes to the potential deployment of robotic guide systems in general use cases, enabling blind users to explore independently.

Chapter 7

Memory-Maze: Scenario Driven Visual Language Navigation Benchmark for Guiding Blind People

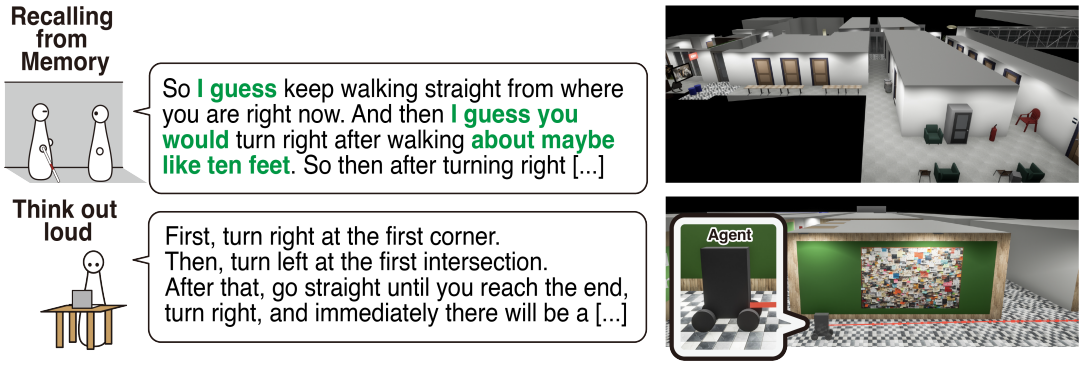


FIGURE 7.1: **Memory-Maze Benchmark.** Left: the instructions obtained in the memory-based scenario contain unique phrases, highlighted in green, in contrast to those collected in traditional think-out-loud settings. Right: Our benchmark environment based on the CARLA simulator [232], and the VLN agent that navigates within the environment.

7.1 Introduction

In this chapter, we again address navigation scenarios with map-less robot systems, as addressed by PathFinder (Chapter 5). In the navigation scenario, how the information to the destination is sourced is an important problem to address. In this regard, PathFinder used route descriptions provided by sighted passersby [107]. Nonetheless, in the PathFinder's experiment, participants still asked for the route multiple times and even mistook the route, leading to task failure. This led us to the idea of whether the robot could instead determine its way based directly on the route description given by sighted passersby. Specifically, this scenario, in which the user navigates to the destination based on language instructions [127], corresponds to the same task as Visual Language Navigation (VLN), which has been actively studied in the robotics and computer vision fields. We therefore believe that the PathFinder scenario can be formulated as a VLN task, and that a VLN model could be used to operate the robot instead of relying on the user.

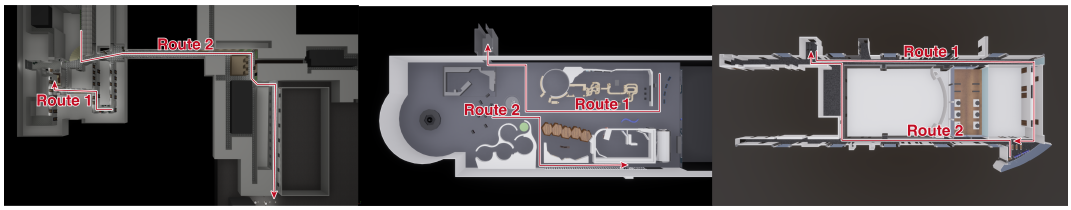
However, direct application of existing VLN models to the blind people navigation scenario is currently limited, as there is a need for a benchmark that reflects the blind users' demands realistically. Many VLN tasks have been addressed in environments such as static houses [127] or roadways [131]. Nonetheless, it is also most important for blind individuals to navigate large public spaces such as shopping malls or university hallways. Compared to existing environments, these environments are characterized by physical turning points and intersections, resembling a maze. Besides the environmental difference, in existing VLN literature, natural language instructions are provided by thinking out loud. In other words, annotators visually navigate a virtual environment and type out instructions for constructing routes concurrently. In our scenario, sighted passersby must describe the route from their memory, which often contains errors such as inaccurate estimates of distances, hallucinations of landmark objects, and omissions of key turning points. To the best of our knowledge, our benchmark is the first to address the scenario of a blind user seeking memory-based instructions from sighted passersby in maze-like public spaces.

We present *Memory-Maze* (Figure 7.1), a benchmark that reflects the blind user navigation scenario. Memory-Maze contains virtual environments of real-world public spaces. It is based on CARLA [232], which enables us to simulate various sensor data (*e.g.*, LiDAR) from robots. It also contains instructions data gathered from two studies from sighted individuals. In the first study, instructions were gathered through online questionnaires by observing walk-through videos from a first-person perspective. This is similar to the annotation method used in existing research. In the second study, instructions were collected in-person by asking sighted passersby to describe the same routes from their memories. This reflects the novel scenarios envisioned in our benchmark. We observed different characteristics among the two studies in terms of length, number of errors, variety, etc.

To analyze the difficulty of our benchmark, we developed a VLN baseline model better designed to navigate in large public spaces, by leveraging modular APIs to handle navigation control and perceptions. Our model also fulfills two requirements for the practical deployment of VLN models for blind people: zero-shot transfer to unseen environments without navigation graphs and single inference. Navigation robots need to be used in unseen environments for blind people, directly applying existing supervised models poses a challenge due to their limited performance in unseen settings [133]. Additionally, existing models perform repeated iterative inferences during navigation, resulting in frequent stops and prolonged navigation time. Leveraging large language models' (LLM) potential for zero-shot generalization in unseen environments, our single-inference LLM-powered model converts the instruction into Python code based on the defined robot control API (Section 7.3.2) for route navigation. This code generation approach modularizes low-level commands such as path-planning for collision avoidance and intersection detection, and serves as a baseline that focuses more on the language interpretation and reasoning capabilities of VLN. Through the study with our model and the current state-of-the-art methods [134, 138], we demonstrated the difficulty of our benchmark and a tendency that real-world memory-based instructions are more difficult for VLN models to handle.

We summarize our contributions below.

1. We constructed Memory-Maze, a benchmark containing virtual environments of a large public spaces, and gathered two sets of instructions, one collected by thinking out loud and one obtained from human memory.



(A) University (B) Museum 5F (C) Museum 7F

FIGURE 7.2: **Bird's-Eye Views of Memory-Maze.** The benchmark contains three environments. The university includes features such as classrooms, offices, hallways, a kitchen, and a library. The 5th floor of the museum mainly contains exhibits. The 7th floor contains conference rooms, hallways, and a terrace area. Each environment includes two routes, totaling six routes. In the on-site study, participants were asked to describe the route from the starting point to the end point, thus, their descriptions may vary from the visualized route.

2. Through an experiment with current state-of-the-art models and our baseline VLN model, we revealed the gap between the instructions collected based on memory and those collected by thinking out loud.

Our benchmark and codes are available at https://github.com/chestnutforestlabo/MemoryMaze.

7.2 Related Work

In this section, we describe the VLN benchmark and models, building on the literature reviewed in Section 2.6.

7.2.1 Benchmarks in VLN tasks

The VLN task has been conducted in various benchmarks, ranging from indoor [127, 130] to outdoor [131, 132] settings. Most of the instruction annotations of these benchmarks were created by annotators who typed while concurrently observing a virtual environment or by researchers who constructed them manually. This way of obtaining instructions is not suitable for our purpose, as it does not reflect the scenario of people describing routes from their memories. Researchers have also explored benchmarks with longer routes for long-horizon navigation tasks [129]. Still, existing benchmarks do not feature large public areas where blind people navigate, such as shopping malls or university hallways. These areas contain both static and dynamic obstacles and are characterized by the existence of turning points and intersections (Figure 7.2). A related benchmark is Touchdown [131], which also emphasizes navigation through intersections and dynamic environments. However, its map structure is represented by a navigation graph (*i.e.*, an undirected graph that represents navigable points with nodes), whereas the Memory-Maze benchmark scenario assumes no prior knowledge such as navigation graphs.

7.2.2 VLN Models

Researchers have explored solutions for VLN tasks using supervised models [127, 128], which learn from a sequence of observations and actions to take. These supervised models often do not transfer well in unseen environments [133]. With the recent advancements in LLM, researchers have also explored methods that do not require retraining [134, 135, 136]. One such approach was to use LLMs to extract

landmarks from instructions and follow chronologically [135]. Another approach was to utilize LLM to flexibly determine actions at each step. NavGPT [134] is a model that uses LLM iteratively to select the node to navigate to within a navigation graph. Additionally, researchers have explored approaches that utilize the code generation capability of LLM [139, 140]. In the method proposed by Biggie *et al.* [140], given a prebuilt 3D map, images from their robot, and a Python API, the model generates codes that locate a target object [233], maps the object's location on the 3D map, and navigates to the mapped location [140]. While these methods are effective when the given instructions include sufficient landmarks, instructions recalled from memory often contain insufficient landmarks, potentially leading to failure. Furthermore, these methods are limited by the need for a navigation graph or 3D map, which is difficult to construct for every unseen environment. To eliminate this requirement, models have been proposed to predict navigation graphs [137] or low-level actions [138] iteratively. However, the need for iterative inference prolongs inference time, which may affect navigation by not reacting to dynamic obstacles responsively. Moreover, iterative inference may be impractical in large public spaces, where the model could be required to perform over many inference steps, due to the need to process long time-horizon data. Our model utilizes LLM to produce navigation codes that follow a specified path in a single iteration, and allows flexible integration of low-level planning algorithms for obstacle avoidance. This direct generation of navigation codes, coupled with existing low-level planning algorithms, allows operation without the need for navigation graphs.

7.3 Memory-Maze

Here, we describe our benchmark's virtual environment and the robot simulation program. To simulate our scenario, we selected a floor of a university building and two floors in a museum building (Figure 7.2), which is characterized by the existence of multiple turning points.

7.3.1 Selecting and Building the Simulator

To simulate a scenario where a robot guides a blind person, it is necessary to simulate high-fidelity egocentric visuals that are realistic enough to run an image recognition algorithm. Thus, we built a novel virtual environment from scratch on top of the CARLA [232] simulator. While primarily developed for autonomous driving simulations, CARLA's flexibility and compatibility with the Unreal Engine allowed us to create a detailed 3D model of the experimental site. CARLA also offers the ability to configure the existence of static and dynamic obstacles and to simulate various sensors like RGB cameras, depth sensors, and LiDAR sensors. We created a 3D model of the experimental site using Fusion 360 and imported it into CARLA. This 3D model accurately reproduces the experimental site, both visually and in terms of floor layout. It also includes major objects along the route (doors, chairs, a statue, *etc*).

7.3.2 Implementation of the Control Program

Our next step was to develop a control program for the robot in the simulator to be used by our baseline VLN model. Utilizing CARLA's Python API to control the navigation robot, we implemented various control functions. We describe four major functions implemented.

We implemented functions for the agent to move forward (`move_forward(distance)`), find a turning point (`detect_turning_point()`), and turn (`turn(direction)`) using CARLA's `vehicle.apply_control` API. When using the `move_forward(distance)` function, to ensure the robot moves along the path without colliding with walls, we implemented a feature that makes the robot navigate as closely to the center of the corridor as possible. We calculate the central path based on the coordinates of the four corners of the corridor in the 3D model. The central path tracking is realized through PID control, which adjusts the robot's steering angles. When the `detect_turning_point()` function is used, it determines if the robot is in the pre-annotated areas of turning points and returns navigable directions if the robot is in one of them. Once the robot is at the turning point, it could change its direction using the `turn(direction)` function. Because component algorithm development of the control program was beyond the scope of this study, in our experiment, coordinates of the corridor's corners and the turning point areas are acquired from the virtual environment, reducing errors from noise in perception or control, and focusing on executing instructions. However, these can be obtained using prior well-established methods [24].

Additionally, we implemented an image recognition module `detect_from_RGB_image(object)`, which outputs bounding boxes of all detected objects, to manage landmark-related instructions such as *"turn after finding a chair."* While most existing object detection models are designed to identify objects from predefined classes, they are not capable of detecting arbitrary objects. Therefore, we used Grounding DINO [234], an open-vocabulary object detection model. Open-vocabulary object detection models output bounding boxes for any object by using the object's name as a query. With the object detection model selected, we then used CARLA's robot ego-centric RGB sensors to capture images. To address tasks requiring the robot to identify an object multiple times (*e.g.*, *"turn after passing four doors"*), we added tracking algorithms to avoid counting the same object in different frames as distinct entities. We further assume that in instructions that require finding landmark objects, the objects are located in close vicinity. For example, in the instruction *"turn after finding [object],"* although the camera could capture the object at a considerable distance, such instructions typically imply that *"[object]"* is near the robot. Therefore, we utilized the depth sensors available in CARLA to measure the distance to each object in the image, filtering out objects that are far away to ensure only those at close range are detected. We set the distance threshold to be four meters.

7.4 Instruction Data Collection

7.4.1 Procedure

We conducted two studies, one online and one onsite, to collect natural language instruction data for routes at three locations: a floor across three buildings in a university and two floors in a museum. We designed the route as shown in Figure 7.2. The studies were approved by our institutional review board (IRB), and informed consent was obtained from all participants. For each route, we obtained two rounds of instructions: one asking participants to describe the route to a blind person with a navigation robot naturally (first iteration) and another asking participants to describe the route after providing them with a brief description of the capability of the navigation robot (second iteration). The second instruction was collected to obtain more accurate memory-based instructions given by passersby. This was achieved by explaining the robot's capability (*e.g.*, being able to detect objects) to the participants.

It simulates a scenario where a blind user may provide sighted passersby with robot information to obtain refined instructions. We expect that telling them about robots' capabilities would enable VLN models to achieve better performance.

In the first study, participants completed an online questionnaire designed to gather instructions that were similar to those in prior works. They were first presented with a scenario in which they communicated with a blind person accompanied by a navigation robot capable of following natural language instructions and 360° video walkthroughs of two routes. They were then asked to type instructions to the destination. They were allowed to re-watch the walkthrough videos at any time. We collected four instructions per participant. In total, 78 participants participated in the study, resulting in 312 instructions. The participants were gathered through university recruitment or through an online survey platform, and all were unfamiliar with the shown routes. The study was conducted in Japan, and the instructions were translated into English using GPT-4.

In the second study, we conducted an onsite in-person study. The aim of this study was to gather data that reflects the realistic scenario of sighted people describing the route from their memory. Thus, they did not watch the walkthrough video or experience the route during the study. The experimenter roleplayed as blind individuals, asked them for directions to the route destinations, and instructed them to describe the route verbally in two rounds. For the first iteration, we asked participants to describe the routes as naturally as possible. For the second iteration, to obtain more accurate instructions for the benchmark, in addition to explaining the robot's capabilities, the experimenter pointed out errors in the participants' given instructions, such as a missing turn, and asked them to explain the route again. For the university routes, we recruited sighted passersby and ensured that all participants were familiar with the route by using a pre-study check survey. In this study, each participant described a single route, resulting in two instructions per participant. In total, 40 participants participated in the study at the university, contributing 80 instructions. For the museum routes, we recruited staff or recent visitors who were familiar with the museum layouts. In this study, each participant described two routes, resulting in four instructions per participant. In total, 43 participants participated in the study at the museum, contributing 172 instructions.

7.4.2 Benchmark Analysis and Statistics

The mean, median, and standard deviation (SD) for the length of collected instructions are reported in Table 7.1. First, we observed that the mean length and SD are longer for the second iteration in most cases, as participants tended to add more information on the second iteration. Also, we observed a tendency for instructions collected onsite to have higher lengths and more length variation. This is because, in the online study, participants described relevant and mostly accurate information about landmarks and turning points, while in the onsite study, many participants tried to be descriptive, relying on their memory, such as adding audio, olfactory cues, and conversational phrases such as *"I'm not 100% sure about this, but I think..."*.

The average instruction length and route distance in our benchmark are greater than those in previous datasets. For example, the R2R dataset includes instructions averaging approximately 30 words and route distances of about 10 meters [127], and the RxR dataset features instructions averaging 78 words and route distances of 14.9 meters [235].

The word clouds of the collected instructions are shown in Figure 7.3. For the university environment, although the samples collected in the onsite study are fewer,

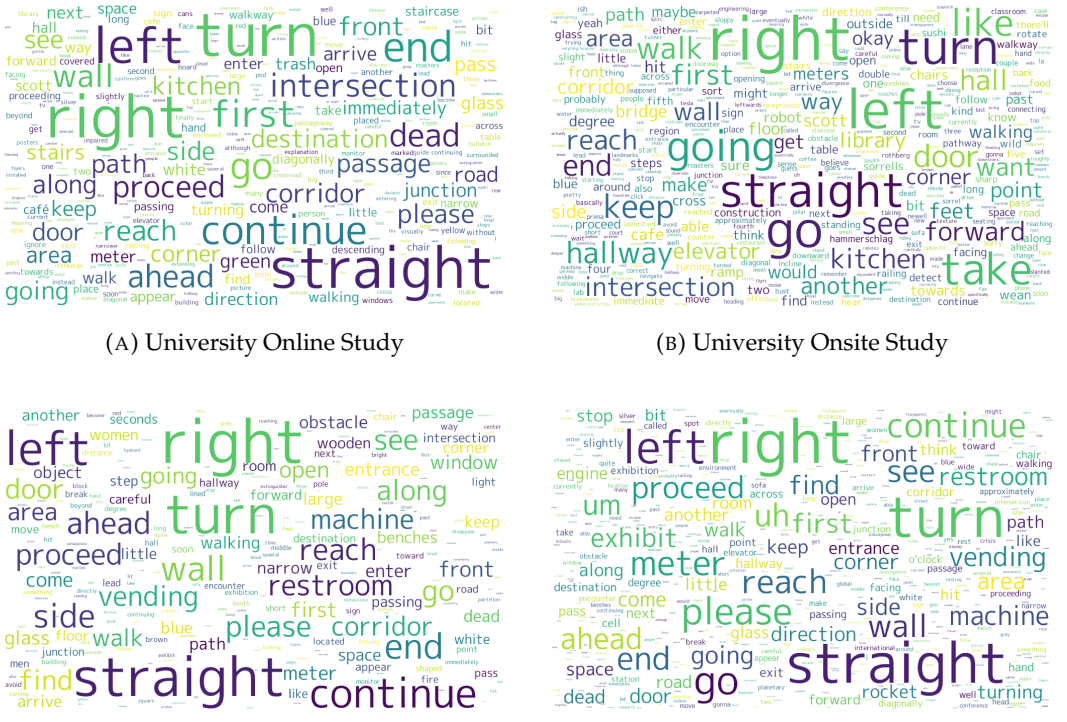
TABLE 7.1: **Data Analysis.** The table presents the route length (RL), mean, median, and standard deviation (SD) of word counts in the collected instructions, and their failure rates (FR). For the onsite instructions, we also report the alternative rate (AR), the rate of describing alternative routes.

	Route	RL	Iteration	Mean	Median	SD	FR	AR
Online Study	University R1	40.27m	1	51.8	47.0	17.8	0.0%	-
			2	69.8	64.0	19.9	0.0%	-
	University R2	156.68m	1	81.3	81.0	24.9	9.09%	-
			2	98.3	99.0	31.2	3.03%	-
	Museum 5F R1	71.18m	1	81.4	78.0	36.3	17.39%	-
			2	88.9	90.0	32.3	17.39%	-
	Museum 5F R2	44.05m	1	60.1	53.5	21.5	9.09%	-
			2	71.0	61.0	33.9	4.55%	-
	Museum 7F R1	86.10m	1	98.2	91.5	42.5	13.64%	-
			2	96.7	90.0	42.2	18.18%	-
	Museum 7F R2	79.40m	1	71.3	68.0	25.8	4.35%	-
			2	95.0	85.0	47.4	0.00%	-
Onsite Study	University R1	40.27m	1	73.9	74.5	36.6	25.0%	10.0%
			2	102.9	94.5	51.1	25.0%	10.0%
	University R2	156.68m	1	131.0	115.5	73.2	40.0%	15.0%
			2	147.3	143.0	65.0	35.0%	15.0%
	Museum 5F R1	71.18m	1	68.2	64.0	27.4	76.19%	0.0%
			2	97.1	92.0	27.0	9.52%	0.0%
	Museum 5F R2	44.05m	1	65.5	51.0	42.7	45.45%	59.1%
			2	83.4	68.5	39.7	4.55%	63.6%
	Museum 7F R1	86.10m	1	68.7	69.5	27.5	54.54%	4.5%
			2	89.0	84.0	24.0	13.64%	9.0%
	Museum 7F R2	79.40m	1	79.5	69.0	40.0	52.38%	85.7%
			2	99.0	96.0	37.5	23.81%	90.1%

they include 521 different words compared to the 381 words found in the samples from the online study. The same trend was noted in the museum environment, with 611 different words found in the onsite study and 586 words in the online study. This shows the greater diversity in the instructions' wording when described from memory. Although the instructions from the online study were translated using LLM, we believe that these results hold in the instructions' original language.

In Table 7.1, we also manually analyzed each instruction to determine if it contained significant errors, *i.e.*, the number of failures in describing the route correctly. One author first conducted an initial failure review, after which multiple authors engaged in a discussion to reach a consensus on all samples. Instructions in the online study were classified as failures for reasons such as turning in the wrong direction; instructing a turn at the incorrect turning point; and suggesting unnecessary extra turns. For instructions collected in the onsite study, the reasons for the failures were containing extra turns, directing to an incorrect direction, leading to a wrong destination, lacking essential turn information, turning to incorrect directions at a turn, and containing inaccurate environmental details.

Interestingly, while examining the instructions, we realized that in the real world, humans may be able to correct errors in the instructions. For example, according to some passersby, the robot should go through a corridor between the hexagon



(A) University Online Study (B) University Onsite Study

(C) Museum Online Study (D) Museum Onsite Study

FIGURE 7.3: **Word Clouds.** The onsite instruction data contains unique phrases that come from talking while recalling from the memory, such as *"uh," "maybe,"* and *"okay."*

exhibitions and the rectangular exhibition immediately to the right for museum 5F R2. However, there is actually no corridor between these two exhibitions. But it is possible to imagine where the nonexistent corridor might lead and try to find a detour. Being able to recognize errors in memory-based instructions is vital for aiding blind people to follow instructions provided by passersby. This is a unique aspect of our benchmark.

As shown in Table 7.1, during the onsite study, we observed that participants sometimes described alternative routes compared to those we had initially anticipated and illustrated in Figure 7.2, when they described the routes from their memory. Surprisingly, some participants described a different route in the second iteration compared to their initial description. This highlights the potential of humans to describe alternative routes in real-world scenarios and the need for VLN models to perform equally well in these alternative routes, thereby underscoring the naturalness of our benchmark.

In summary, our benchmark captures real-world complexity beyond mere geometric turns. We found that even routes with 90° turns are accompanied by richly varied natural language instructions. For example, ∼25% of participants described the two consecutive turns near the goal at Museum 5F R2 as *"proceed in the front right direction"*. In another example, one instruction described the consecutive turns in University R2 with only semantic cues such as *"[...], I believe and then you walk along the hallway connecting DEF hall and the GHI hall and then you'll arrive at your destination."* Memory-Maze provides a unique scenario-based evaluative environment by incorporating variation in intersection counts, route lengths, and landmark complexity.

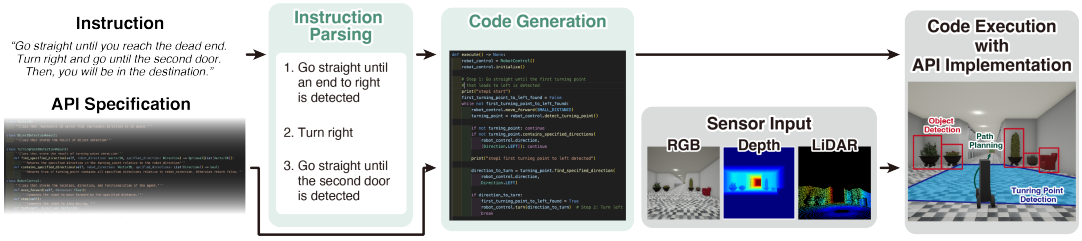


FIGURE 7.4: **Method Overview.** Given a set of instructions from a sighted passerby, the LLM first parses it into an itemized format. Then, combined with the API specification, the LLM generates Python code directly to control the robot, which runs in the virtual environment using the simulated sensor inputs.

7.5 Baseline VLN Model Implementation

Our baseline VLN model uses the control API specification from our benchmark so that we may focus more on its language interpretation and reasoning capabilities. First, we utilized LLM's capability to generalize to various tasks and comprehend complex natural language instructions, so that no additional training is required when deployed in a new environment. Second, our method requires only a single inference iteration to generate low-level navigation code for robot control, in contrast to existing models that perform multiple inferences during navigation, which may prolong navigation time. It also eliminates the need for a navigation graph by generating codes that directly interface with low-level navigation modules. The generation of navigation code potentially leads to the flexibility of integrating existing, well-established methods into various modules, such as for obstacle avoidance [11] or turning point detection [24, 25].

We define the following as inputs to the agent: *natural language instruction*, the *sensor input* which includes the details obtained from sensors, *API specification* which consists of the commands and their explanations in Python that the agent can use as described in Section 7.3.2, *API implementation* which is the actual implementation of the API specification, and the *initial orientation* of the robot. We assume the initial orientation is predetermined, as the blind user can adjust it in place. We used the GPT-4 (gpt-4-1106-preview) model for the LLM. For the initial setup of the prompt, instructions from five participants in the online study were used as references to construct the prompt for the proposed systems. Figure 7.4 shows the implementation overview.

7.5.1 Parsing Instruction

The system first parses a natural language instruction to step-by-step instructions using LLM. This was done to organize our benchmark's diverse natural language instruction and make it more interpretable before generating navigation codes. To achieve this, we prompt LLMs with a set of rules they should follow, such as the requirement to describe when and which turning point to turn, and which object the robot should detect, along with examples of possible input and expected output. After parsing, each navigation step is returned as a brief sentence. We employ a two-stage prompting method to guide LLM for more accurate outputs. We prompt LLM to provide a thought to guide the generation of the first output, then refine the output by incorporating a second thought, leading to the finalized output.

7.5.2 Navigation Code Generation

To generate the navigation code, we prompt LLM with an API specification that includes a range of commands for robot operations (*e.g.*, `move_forward(distance)` function). These commands are complete with docstrings of their usage explanation [140, 233] and instructions to generate Python codes that follow the provided specification. For `detect_from_RGB_image(object)`, our model uses an open vocabulary object detector internally (Section 7.3.2) and flexibly determines which object to detect by generating an object argument. For example, for an input requesting the location of a red chair, the function would be invoked as: `detect_from_RGB_image("red chair")` by an LLM while generating a code. The API specification was formatted to the similar format of the previous work [140, 233], but with additional notes, such as how each function should be and not be used. We again employ the same two-stage prompting method. Finally, we execute the generated code using the API implementation.

7.6 Experiment

Our benchmark simulates a scenario in which blind people ask sighted passersby to provide route guidance [24] from their memories. To evaluate how current models perform under this setting, we conducted an experiment.

7.6.1 State-of-the-Art Models

In our scenario, VLN agents are expected to demonstrate strong transferability, as blind users may navigate across diverse unseen locations by asking sighted passersby for directions. To evaluate this capability, we compare our model with two prior state-of-the-art methods that leverage foundation models and exhibit strong zero-shot performance: NavGPT [134] and NaVid [138].

NavGPT [134] demonstrates strong zero-shot transfer capability by leveraging an LLM, visual foundation model, and an object detector to iteratively select destinations within a navigation graph until the agent determines it has reached the goal. We used GPT-4o-mini for the LLM and for the visual foundation model, and the same Grounding DINO [234] for the object detector. As NavGPT requires a navigation graph to operate, we constructed navigation graphs over the environments following the R2R dataset [127].

NaVid [138], a state-of-the-art VLN model that demonstrates strong generalization to unseen environments, employs a visual foundation model and operates without a navigation graph, relying solely on camera input, similar to our model. We strictly controlled the agent by following NaVid's established pipeline. We initialized its weights of LLM (Vicuna-7B) using their open-sourced checkpoint [138].

7.6.2 Metrics

For the metrics, we employ success rate (SR), oracle success rate (OSR), and shortest path distance (SPD) [127, 236], and coverage weighted by length score (CLS) [237, 238]. As CLS computes the similarity of the path on the graph, it requires a dense navigation graph to map the navigated trajectory onto. Thus, we divided passable corridors into 50 cm square grids to serve as nodes on a graph and mapped predicted

and ground truth paths onto it to calculate this metric [238]. For routes where participants described an alternative path, we used the described route as the ground truth.

7.7 Results and Discussion

TABLE 7.2: **Performance of VLN Models.** We compare our method with state-of-the-art VLN models that fulfill the requirements relevant to our scenario.

Condition			Online Study Data				Onsite Study Data			
Method	Parser	Route	SR↑	OSR↑	SPD↓	CLS↑	SR↑	OSR↑	SPD↓	CLS↑
NavGPT		University R1	0.04	0.09	37.54	0.05	0.02	0.04	40.58	0.05
NaVid		University R1	0.00	0.00	35.73	0.03	0.00	0.00	36.67	0.02
Proposed		University R1	0.20	0.24	**17.01**	**0.56**	0.23	0.33	21.04	0.46
Proposed	✓	University R1	**0.30**	**0.35**	19.66	0.49	**0.30**	**0.38**	**18.32**	**0.54**
NavGPT		University R2	0.00	0.00	162.10	0.01	0.00	0.00	161.13	0.01
NaVid		University R2	0.00	0.00	149.79	0.00	0.00	0.00	151.47	0.00
Proposed		University R2	0.00	0.00	93.66	0.32	**0.03**	**0.03**	117.52	0.20
Proposed	✓	University R2	**0.04**	**0.04**	**81.59**	**0.38**	**0.03**	**0.03**	**98.13**	**0.29**
NavGPT		Museum 5F R1	0.00	0.00	50.76	0.00	0.00	0.00	51.30	0.00
NaVid		Museum 5F R1	0.00	0.00	54.59	0.01	0.00	0.00	55.50	0.01
Proposed		Museum 5F R1	0.11	0.20	35.46	0.44	0.02	0.02	43.35	0.32
Proposed	✓	Museum 5F R1	**0.20**	**0.26**	**26.71**	**0.60**	**0.05**	**0.07**	**29.14**	**0.54**
NavGPT		Museum 5F R2	0.00	0.07	37.47	0.07	0.00	0.07	35.67	0.06
NaVid		Museum 5F R2	0.00	0.00	43.74	0.01	0.00	0.00	43.82	0.01
Proposed		Museum 5F R2	**0.05**	0.18	23.17	0.25	0.00	**0.02**	29.08	**0.37**
Proposed	✓	Museum 5F R2	**0.05**	**0.32**	**16.43**	**0.29**	0.00	**0.02**	**24.78**	**0.37**
NavGPT		Museum 7F R1	0.00	0.00	54.70	0.08	0.00	0.00	36.00	0.14
NaVid		Museum 7F R1	0.00	0.00	73.76	0.06	0.00	0.00	71.30	0.07
Proposed		Museum 7F R1	**0.02**	**0.02**	55.99	0.31	0.05	0.05	46.67	0.42
Proposed	✓	Museum 7F R1	**0.02**	**0.02**	**42.59**	**0.48**	**0.09**	**0.09**	**25.22**	**0.67**
NavGPT		Museum 7F R2	0.00	0.00	61.64	0.01	0.00	0.00	60.43	0.01
NaVid		Museum 7F R2	0.00	0.00	67.39	0.02	0.00	0.00	69.18	0.00
Proposed		Museum 7F R2	**0.15**	0.26	47.41	0.17	0.00	0.10	52.81	0.16
Proposed	✓	Museum 7F R2	0.07	**0.35**	**36.78**	**0.25**	**0.02**	**0.12**	**46.74**	**0.22**

Table 7.2 reports the results of the study.

7.7.1 Performance of the Proposed Method

As shown in Table 7.2, our model outperforms NavGPT and NaVid. The baselines' suboptimal performances can be attributed to two factors: their deviation from the correct direction, and their premature decision that they had reached the goal. This is due to the fact that the baselines refer to the environment at each navigation step with an LLM. For example, if NavGPT makes a mistake even once during this process, it will be challenging for the model to recover the agent back to the correct path. Additionally, NaVid tends to make unnecessary frequent turns after initially following the route correctly for several iterations, likely due to the longer sequence of turns and longer instructions in our benchmark, which NaVid was not trained to handle. In contrast, our method achieves the desired outcome through a single iteration of code generation inference, removing the need to initiate inferences at every intermediate step for instructions like *"go straight for 100m and then turn right."*. We

also observed that the instruction parsing module boosted the performance of our method in most cases.

7.7.2 Difficulty of the Benchmark

In Table 7.2, it is observed that the performances from onsite memory-based instructions tended to be lower than those from online think-out-loud instructions, as it is more likely for the route instructions to contain errors due to human memory, and it is harder for the system to recover from errors. Overall, our results demonstrate the difficulty of the instruction data from human memory and the value of our benchmark.

Across all routes, both our model and the baselines showed suboptimal or low performance. One major reason was the difficulty in handling the varied and inaccurate input instructions. In longer routes, participants tended to inaccurately estimate lengths for certain path segments and not include sufficient information about the destination. Also, because our baseline contained modularized perception and control modules to focus on language parsing and reasoning capabilities, its suboptimal performance implies that the primary challenge in our benchmark lies in the complexity of the language instructions, such as inaccuracies or variations in wording, which were not present in previous benchmarks. Upon closer inspection, many instructions in our benchmark contained phrases that required a combined understanding of both natural language and the building's structure, which our proposed model failed to follow. One example was a phrase such as *"go along this path and turn right in the first intersection,"* which was often described at the starting point of University R1. The instruction skips the right turn in the first turning point by describing it as *"go along this path,"* because the building structure only allows a right turn at the immediate corner. As a result, the instruction starts by describing the first left turn where there are two possible directions to proceed. This variation in the levels of topological details further highlights the difficulty of our benchmark, which imitates real-world scenarios of blind people seeking navigation instructions.

Furthermore, we realized that in the real-world, humans may be able to correct errors in the instructions. For example, some passersby instructed the robot to take a nonexistent corridor between the hexagon and the rectangular exhibitions near the museum 5F R2. Although the corridor did not exist, imagining its intended destination allowed locating an alternative route. Similarly, when participants provided incorrect turns, landmarks described later in the instructions helped identify and correct these oversights.

7.7.3 Effect of Refining Instruction

Table 7.3, reports the performance of our proposed method across different instruction iterations. The first iteration corresponds to the most natural, memory-based instruction, while the second represents memory-based instructions that are more accurate and contain features that may better assist the VLN agent in navigation. Generally, we found that refining the instruction led to a slight performance improvement. This suggests that, when deploying VLN-equipped robots, it is beneficial to assist sighted passersby in recalling routes more and in conveying environmental information in a format that is more compatible with robotic interpretation. However, for Museum 5F R1 and University R1, the tendency was not always the case. This happened because participants tended to describe more objects during the second iteration, which contained greater variation in object descriptions. For

TABLE 7.3: **Effect of Instruction Refinement.** While in most cases refining instruction leads to an increase in performance, in certain cases, it was not always the case, due to redundant referral to surrounding objects.

Condition		Online Study Data				Onsite Study Data			
Route	Iteration	SR↑	OSR↑	SPD↓	CLS↑	SR↑	OSR↑	SPD↓	CLS↑
University R1	1	**0.43**	**0.48**	**16.37**	**0.56**	0.25	**0.40**	20.41	0.52
University R1	2	0.17	0.22	22.94	0.41	**0.35**	0.35	**16.23**	**0.56**
University R2	1	**0.04**	**0.04**	86.82	0.36	0.00	0.00	**93.32**	**0.31**
University R2	2	**0.04**	**0.04**	**76.36**	**0.41**	**0.05**	**0.05**	102.95	0.28
Museum 5F R1	1	**0.26**	**0.30**	**25.80**	0.60	0.00	0.00	**27.21**	**0.56**
Museum 5F R1	2	0.13	0.22	27.61	**0.61**	**0.10**	**0.14**	31.07	0.52
Museum 5F R2	1	**0.05**	0.27	17.77	0.27	0.00	0.00	25.75	0.32
Museum 5F R2	2	**0.05**	**0.36**	**15.09**	**0.30**	0.00	**0.05**	**23.82**	**0.43**
Museum 7F R1	1	0.00	0.00	46.57	0.43	**0.09**	**0.09**	**23.77**	**0.68**
Museum 7F R1	2	**0.05**	**0.05**	**38.61**	**0.53**	**0.09**	**0.09**	26.66	0.65
Museum 7F R2	1	0.00	0.26	37.24	**0.26**	0.00	0.05	**46.71**	0.21
Museum 7F R2	2	**0.13**	**0.43**	**36.33**	0.24	**0.05**	**0.19**	46.76	**0.23**

example, one participant described only turning point-related information at the first iteration, while in the second iteration, the participant also described objects to ignore, such as, *"then, you will come across an intersection with a door on the left and an intersection with doors on both sides, but ignore them and continue straight ahead."*

7.8 Conclusion and Future Work

This work proposed Memory-Maze. We found that realistic instructions collected in the onsite environment, where participants had to rely on human memories, were longer with greater variation in words, and contained more errors compared to the instructions collected online. Upon qualitative inspection, we observed evidence of the tendency for memory-based instruction to be more difficult for the model to handle, such as the ones that required understanding of *"go along this path."* This suggests that future VLN models should consider a more adaptive map representation where nodes and turns are not strictly defined, or a more flexible approach to accommodate varying topological descriptions.

Our central finding is that the current VLN model alone may still be insufficient for completing this task. Therefore, for future work, we aim to explore the interactive aspects between users and robots. For example, we plan to investigate shared control with a VLN model, in which the robot usually takes over control and the user intervenes when needed; as proposed by Chi *et al.* [239], the robot can ask the user when it is uncertain about the way, and the user can intervene in such cases. Another solution is that the robot could guide the instruction from passersby to be better or rephrase it by itself, potentially leading to improved performance. We also plan to convert our baseline into a closed-loop architecture that continuously verifies and refines its interpretation of instructions using real-time sensor input.

Lastly, although the size of our benchmark is comparable to existing benchmarks [240, 241], its size remains limited in order to be used as a dataset for training. One possible approach is to modify the design of the online study so that annotators

can only observe the environment prior to providing annotations, but this would only partially replicate the characteristics of real-world memory-based data. Another potential method is to leverage LLMs with in-context learning to augment the benchmark. However, further investigation is needed to ensure that LLM-generated data can accurately mimic the characteristics of our dataset.

Chapter 8

Discussion and Conclusion

8.1 Discussion

We introduced four systems: two smartphone systems, Corridor-Walker (Chapter 3) and Snap&Nav (Chapter 4), and two robot systems, PathFinder (Chapter 5) and WanderGuide (Chapter 6). We also introduced a VLN benchmark, called Memory-Maze (Chapter 7). Among them, four systems or benchmarks (Corridor-Walker, Snap&Nav, PathFinder, and Memory-Maze) targeted the navigation task, *i.e.*, going from one place to another, while one system targeted the exploration task (WanderGuide). Below, we revisit the research questions presented in the introduction section and further describe future opportunities in the area of map-less navigation technology.

8.1.1 Collaborative Interaction (RQ1)

All of our systems adopted collaboration, in which humans are involved in the system's sensing or decision-making process or participate in the system usage flow to support task completion. This collaboration occurs between the system and the human in various forms to achieve shared goals. For example, the collaborative interaction adopted in Corridor-Walker and Snap&Nav was *scanning*, which enabled the user to complement sensing and provide users with more accurate detection results. PathFinder imitated the interaction observed when blind people use guide dogs, in which the aid takes over mobility while the user takes over decision-making. While this guide dog interaction is typically limited to situations where users already know their destination, PathFinder extended this usage to unfamiliar places by conveying information important to wayfinding, specifically sign information.

In Snap&Nav, the collaboration extended beyond the system and the user to include sighted passersby. Sighted passersby captured and verified floor map images for the blind user. The sighted passersby showed willingness to perform this task and even described that this task could be easier than verbally describing routes, which can sometimes be inaccurate. Taken together, we conclude that this dissertation was able to provide an initial design and validate collaborative interaction for assistive navigation technology for blind users.

In terms of interaction paradigms, this opens a new direction in assistive technology, as such systems have typically been designed to be used in a *passive* manner by the end user. Thouse would include autonomous navigation systems such as CaBot [11] or smartphone navigation system such as NavCog [9] and Google Maps [38] How blind people can *actively* take part in navigation thus becomes a new area of research; in particular, participation in the sensing and decision-making process remains both difficult.

In this regard, future work should pursue stronger collaborative interaction. While some tasks presented in this research are automatable, such as navigating through routes described by sighted passersby, there remains room for human involvement. For example, even when we adopt VLN technology for map-less navigation, performance concerns remain, as demonstrated in our findings. One possible approach is to adopt a model that queries users when they are uncertain [239]. Alternatively, the VLN model can serve as a monitor of the human decision-making process, issuing alerts when discrepancies arise between the user's decision and the system's predicted direction of navigation.

Shared control to address challenging scenarios is another direction. An example is presented by Kamikubo *et al.* [89], which addresses scenarios where a robot must navigate through a crowd in which path planning alone cannot resolve navigation due to excessive surrounding people. This situation also occurred in the PathFinder experiment. In such crowded scenarios, one approach is to alert people to the presence of the blind user so that they move [61]. While such a solution, in which the system alerts people, has been presented [61], it is also possible for blind users to take part in that interaction by verbally asking for assistance, based on descriptions obtained from the robot perception. This also connects to the mixed-initiative paradigm, where both the robot and the user take part in the interaction initiative. How to collaboratively navigate through challenging scenarios together with the user and the system remains a topic to explore.

Another opportunity can be observed in the experiment with WanderGuide. Although the robot had a manual mode that allowed users to intervene in the system movement, it did not take users exactly to their intended destination, which is known as the last one meter problem. As users may perceive that they cannot physically reach what they aim for, incorporating functionality that allows users to intervene and adjust the robot position may be a simple yet useful feature. Finally, how to effectively ask sighted humans for help remains another important opportunity for collaboration. While independence is ideal, as demonstrated by Snap&Nav, designs that include sighted people tend to be accepted and can serve as useful information sources.

8.1.2 Comparison with Existing Solution (RQ2)

Compared to regular aids (*e.g.*, white cane), our smartphone navigation systems (Corridor-Walker, and Snap&Nav) showed that they generally required longer task completion time but resulted in increased confidence, lower cognitive load, and fewer collisions. The major portion of the time was spent on scanning, which was designed as a collaborative interaction. Additionally, although the obstacle avoidance function provided the advantage of safer navigation, it introduced additional time because users were required to follow instructions while handling mobility themselves. This is supported by the fact that the smartphone system Corridor-Walker, which included obstacle avoidance, and the smartphone system Snap&Nav, which did not due to system design, showed that Snap&Nav's task completion time did not differ between the cane-only and system-aided conditions. It is also worth noting that we observed users commenting that being able to know the shape of an intersection, even when they did not plan to turn, was not possible with their regular aids.

Showing a similar trend, the robotic system PathFinder, when compared with the map-based topline system CaBot [11], exhibited longer task completion times due to the additional user interaction required, whereas the map-based system did

not require such interaction. Nonetheless, for both smartphone and robotic systems, confidence and cognitive load showed improvement compared to regular aids. For PathFinder, performance was rated between their regular aid and the topline system, and can be concluded to be an "in-between" aid. Some participants further commented that the sense of control they felt through collaborative interaction with the system was positive, as they were sometimes unable to fully trust the map-based topline system. While all systems resulted in longer task completion time, these findings highlight that increased completion time is not necessarily a negative outcome. This also echoes the discussion by Igarashi [242] in the domain of content creation tools, suggesting that performing tasks easily and quickly may have drawbacks, such as reduced richness of experience, and therefore may not always be a goal to achieve. This overall trend indicates that collaborative interaction requires time but also provides advantages. Future work should consider how to reduce interaction time while also balancing task completion time with the benefits that blind users obtain.

8.1.3 Route and Environmental Information (RQ3)

How source route information and what kind of environmental information we conveyed during the task was an important topic. In this dissertation, we assume three cases: the user knows the route (Chapter 3), the user navigates based on route descriptions provided by sighted passersby (Chapter 5 and Chapter 7), and navigation based on captured floor map images (Chapter 4). In the situation where the users were familiar with the route, the intersection information was sufficient to reach the destination. Among these, the easiest method is route description by sighted passersby. When such descriptions are accurate, as shown in PathFinder (Chapter 5), blind users were able to determine their way in combination with intersection detection and sign recognition functionality. However, while users can interpret these descriptions, they remain vulnerable to human error, such as inaccurate route information or instances where a blind user chooses an incorrect direction. Therefore, we extended the interpretation of route descriptions using a VLN model, aiming for the robot to determine the route and to analyze what kinds of errors may occur. As we showed, route descriptions obtained in real-world settings contain more errors than those collected through conventional methods, and route interpretations by robots can still be incorrect. Potential future work includes developing a closed-loop model that interprets instructions while accounting for possible errors, or adopting a hybrid VLN approach as described in Section 8.1.1.

Another source of route information is floor map images, which can be used in certain buildings, coupled with an intersection detection method. While having such information makes the setting closer to map-based navigation, several technical challenges remain. A major challenge lies in analyzing floor map images, as they may vary in scale across locations and may not be well represented by simple topological maps due to open spaces or the lack of information about the user's current location. Localizing and tracking the user's position within the map is another challenge. In our study, we used corridor-like environments, but real-world environments may be more complex and cannot always be represented as simple corridors or perpendicular intersections. This calls for interdisciplinary research, involving more generalized intersection detection technology that can associate node map information even in complex intersections, where the node map is converted from floor map images. It also requires developing floor map analysis algorithms that can adapt to the complexities present in real-world floor maps.

One possible solution, which we did not consider in this dissertation, is to reuse maps constructed during previous visits. For example, users could use WanderGuide to explore a facility once and construct a map of the environment. Reusing this as a route information source would benefit map-less navigation systems, and sharing constructed maps in the cloud would also support scalability. However, this does not convert the problem into a fully map-based setting as in previous accessibility research, because how to construct such maps remains an open research topic. Although both PathFinder and WanderGuide construct maps, PathFinder builds a simple LiDAR map, whereas WanderGuide constructs a semantic map in which captured images are aligned with the LiDAR map. These do not guarantee that the map contains sufficient semantic information for destination-oriented navigation, such as the names and locations of destinations. Research on WanderGuide further highlighted the need for audio information to be embedded into maps and suggested that more fine-grained representations than 2D LiDAR maps, such as 3D reconstruction, may be required to comprehensively explain the environment. How to construct such maps and how to reuse them as route sources remain open topics.

8.1.4 Interdiciplinery Effort

Beyond the field of HCI, this dissertation also extends to VLN research in the robotics field. While it is important to explore the design space of map-less systems, the true realization of such systems also requires progress in each fundamental technical component. Progress in technology from other fields alone does not necessarily resolve these components, as they often involve unique requirements tailored to blind users. This calls for interdisciplinary efforts among accessibility researchers and designers, together with researchers from other fields such as computer vision and robotics.

Specifically, in this dissertation, we developed a VLN benchmark to evaluate system performance in scenarios where blind users navigate based on route descriptions provided by sighted passersby. Another potential effort lies in real-time conversational interaction with MLLMs. For example, one possible reason that the workload was high for WanderGuide may be attributed to the Q&A functionality, which required querying the GPT-4o model through the cloud, resulting in interaction delays. If such interaction could be made real-time, as if blind users were conversing with a sighted companion, they might be able to learn more about the surrounding environment, potentially leading to greater exploration of places they find interesting. With recent progress in mobile MLLMs [243], which enables rapid information access and dialogue, we believe that such real-time interaction will become increasingly feasible. Nonetheless, we note that these off-the-shelf models often fail to fully satisfy the needs of blind users, particularly in situations where real-time conversation requires concise yet sufficiently informative responses. This indicates a need for developing MLLMs specifically optimized for this purpose. We also note that, as seen in WanderGuide, balancing the advantages of AI-generated scene descriptions with the ability to convey more concrete and detailed features remains another open opportunity for interdisciplinary research. As map-less navigation and exploration introduce novel interaction settings, we believe that many technical components must continue to be developed through interdisciplinary efforts to realize such systems in practice.

8.2 Conclusion

Towards scalable assistive navigation technology, this dissertation presented map-less navigation systems for blind people. In this regard, we introduced four systems: two smartphone systems (Corridor-Walker and Snap&Nav) and two robot systems (PathFinder and WanderGuide). The main points raised throughout the individual studies concerned the balance and trade-offs between the performance degradation caused by map-less settings compared to map-based settings, and the advantages of scalability offered by map-less approaches. For example, our study revealed that smartphone navigation systems (Corridor-Walker and Snap&Nav) increased user confidence, reduced cognitive load, and perceived intersection shapes precisely compared to the use of regular aids, despite requiring more time for navigation. Also, compared to map-based systems, the robotic system PathFinder exhibited longer task completion times and lower levels of confidence and cognitive load. Nevertheless, participants appreciated its potential for scalability and controllability. We further showed that the potential of map-less systems extends beyond navigation to exploration, and observed that participants were able to enjoy the exploration, find places they found interesting, and go to those places. Throughout the research, we adopted collaborative interaction in which human capabilities were consistently incorporated, such as scanning, capturing floor maps, making decisions at intersections, and determining where they want to go during exploration. This interaction was adopted to complement the knowledge lack of the system, which comes from the unique map-less setting. Finally, this thesis described a VLN benchmark that emerged from the need for map-less navigation in scenarios where users go to their destination based on route descriptions provided by sighted passersby. We showed how real-world instructions differ from those previously proposed and found that existing models are not yet sufficient for such scenarios.

Finally, in the discussion, we revisited our research questions and outlined possible future directions. The first direction is to advance collaborative interaction and integrate it more deeply into our systems. For example, we could develop an interactive VLN model for determining directions with fewer errors or for monitoring users' navigation progress. We could also design shared control mechanisms to address challenging scenarios for robots, such as navigating through crowds or solving the last one-meter problem, which WanderGuide was unable to address. Another important direction is to explore how map information previously gathered can be reused for future navigation automatically. Converting this into a problem equivalent to map-based navigation is a non-trivial challenge that still requires effort. We also identified several opportunities for interdisciplinary effort. Advancing floor map recognition algorithms for sourcing route information remains necessary. Finally, developing lightweight mobile LLMs tailored for real-time conversation with blind users would greatly benefit our systems.

We believe that the realization and adoption of map-less navigation technology will impact the daily lives of blind people by enabling them to navigate various places independently. In the ultimate future, users will utilize existing solutions in environments where prebuilt maps and infrastructures are available, and switch to map-less navigation systems where such resources are not accessible. Moreover, the use of map-less systems across diverse spaces will potentially make navigation easier as information accumulates over time. By overcoming the rental business model of robots, blind people may potentially own such robots in the future as an alternative third navigation aid.

Bibliography

[1] Hironobu Takagi, Murata Masakyuki, Sato Daisuke, Tanaka Shunya, Yabuuchi Tomohiro, Kayukawa Seita, and Kimura Shunsuke. "Towards the Adoption of Autonomous Navigation Robots for People with Visual Impairments (Translated to English)". In: *Digital Practice Corner, IPSJ*. Japanese paper available at: https://www.ipsj.or.jp/dp/contents/publication/52/S1304-S02.html. 2022.

[2] William R Wiener, Richard L Welsh, and Bruce B Blasch. *Foundations of Orientation and Mobility*. American Foundation for the Blind, 2010.

[3] Nicholas A Giudice. "Navigating Without Vision: Principles of Blind Spatial Cognition". In: *Handbook of behavioral and cognitive geography*. Edward Elgar Publishing, 2018.

[4] Christin Engel, Karin Müller, Angela Constantinescu, Claudia Loitsch, Vanessa Petrausch, Gerhard Weber, and Rainer Stiefelhagen. "Travelling More Independently: A Requirements Analysis for Accessible Journeys to Unknown Buildings for People with Visual Impairments". In: *ASSETS*. 2020.

[5] Sulaiman Khan, Shah Nazir, and Habib Ullah Khan. "Analysis of Navigation Assistants for Blind and Visually Impaired People: A Systematic Review". In: *IEEE Access* (2021).

[6] Kanak Manjari, Madhushi Verma, and Gaurav Singal. "A Survey on Assistive Technology for Visually Impaired". In: *Internet of Things* (2020).

[7] Kuriakose Bineeth, Shrestha Raju, and Sandnes Frode Eika. "Tools and Technologies for Blind and Visually Impaired Navigation Support: A Review". In: *IETE Technical Review* (2020).

[8] Masaki Kuribayashi, Seita Kayukawa, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima. "LineChaser: A Smartphone-Based Navigation System for Blind People to Stand in Lines". In: *CHI*. 2021.

[9] Daisuke Sato, Uran Oh, João Guerreiro, Dragan Ahmetovic, Kakuya Naito, Hironobu Takagi, Kris M Kitani, and Chieko Asakawa. "NavCog3 in the Wild: Large-scale Blind Indoor Navigation Assistant with Semantic Features". In: *TACCESS* (2019).

[10] Bing Li, J Pablo Munoz, Xuejian Rong, Jizhong Xiao, Yingli Tian, and Aries Arditi. "ISANA: Wearable Context-aware Indoor Assistive Navigation with Obstacle Avoidance for the Blind". In: *ECCV*. 2016.

[11] João Guerreiro, Daisuke Sato, Saki Asakawa, Huixu Dong, Kris M Kitani, and Chieko Asakawa. "CaBot: Designing and Evaluating an Autonomous Navigation Robot for Blind People". In: *ASSETS*. 2019.

[12] BlindSquare. *BlindSquare*. Retrieved in November 25, 2024 from https://www.blindsquare.com/. 2024.

[13] Yuka Kaniwa, Masaki Kuribayashi, Seita Kayukawa, Daisuke Sato, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima. "ChitChatGuide: Enabling Exploration in a Shopping Mall for People with Visual Impairments Through Conversational Interaction Using Large Language Models". In: *PACMHCI* (2024).

[14] Catherine Feng, Shiri Azenkot, and Maya Cakmak. "Designing a Robot Guide for Blind People in Indoor Environments". In: *HRI EA*. 2015.

[15] Vinitha Ranganeni, Mike Sinclair, Eyal Ofek, Amos Miller, Jonathan Campbell, Andrey Kolobov, and Edward Cutrell. "Exploring Levels of Control for a Navigation Assistant for Blind Travelers". In: *HRI*. 2023.

[16] Shaojun Cai, Ashwin Ram, Zhengtai Gou, Mohd Alqama Wasim Shaikh, Yu-An Chen, Yingjia Wan, Kotaro Hara, Shengdong Zhao, and David Hsu. "Navigating Real-World Challenges: A Quadruped Robot Guiding System for Visually Impaired People in Diverse Environments". In: *CHI*. 2024.

[17] Hironobu Takagi, Kakuya Naito, Daisuke Sato, Masayuki Murata, Seita Kayukawa, and Chieko Asakawa. "Field Trials of Autonomous Navigation Robot for Visually Impaired People". In: *CHI EA*. 2025.

[18] Seita Kayukawa, Daisuke Sato, Masayuki Murata, Tatsuya Ishihara, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. "Enhancing Blind Visitor's Autonomy in a Science Museum Using an Autonomous Navigation Robot". In: *CHI*. 2023.

[19] Miraikan. *About the "AI Suitcase"*. Retrieved in December 20, 2025 from https://www.miraikan.jst.go.jp/en/lab/AIsuitcase. 2025.

[20] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, and Eelke Folmer. "The User as a Sensor: Navigating Users with Visual Impairments in Indoor Spaces Using Tactile Landmarks". In: *CHI*. 2012.

[21] Hochul Hwang, Hee-Tae Jung, Nicholas A Giudice, Joydeep Biswas, Sunghoon Ivan Lee, and Donghyun Kim. "Towards Robotic Companions: Understanding Handler-guide Dog Interactions for Informed Guide Dog Robot Design". In: *CHI*. 2024.

[22] Masaki Kuribayashi, Seita Kayukawa, Jayakorn Vongkulbhisal, Chieko Asakawa, Daisuke Sato, Hironobu Takagi, and Shigeo Morishima. "Corridor-Walker: Mobile Indoor Walking Assistance for Blind People to Avoid Obstacles and Recognize Intersections". In: *PACMHCI* (2022).

[23] Masaya Kubota, Masaki Kuribayashi, Seita Kayukawa, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima. "Snap&Nav: Smartphone-based Indoor Navigation System For Blind People Via Floor Map Analysis and Intersection Detection". In: *PACMHCI* (2024).

[24] Masaki Kuribayashi, Tatsuya Ishihara, Daisuke Sato, Jayakorn Vongkulbhisal, Karnik Ram, Seita Kayukawa, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. "PathFinder: Designing a Map-less Navigation System for Blind People in Unfamiliar Buildings". In: *CHI*. 2023.

[25] Masaki Kuribayashi, Kohei Uehara, Allan Wang, Shigeo Morishima, and Chieko Asakawa. "WanderGuide: Indoor Map-less Robotic Guide for Exploration by Blind People". In: *CHI*. 2025.

[26] Masaki Kuribayashi, Kohei Uehara, Allan Wang, Daisuke Sato, Renato Ribeiro, Simon Chu, and Shigeo Morishima. “Memory-Maze: Scenario Driven Visual Language Navigation Benchmark for Guiding Blind People”. In: *RA-L* (2025).

[27] Abdulrhman Alkhanifer and S. Ludi. “Disorientation Factors That Affect the Situation Awareness of the Visually Impaired Individuals in Unfamiliar Indoor Environments”. In: *UAHCI*. 2015.

[28] Watthanasak Jeamwatthanachai, M. Wald, and G. Wills. “Indoor Navigation by Blind People: Behaviors and Challenges in Unfamiliar Spaces and Buildings”. In: *BJVI* (2019).

[29] Victor R Schinazi, Tyler Thrash, and Daniel-Robert Chebat. “Spatial Navigation by Congenitally Blind Individuals”. In: *Wiley Interdisciplinary Reviews: Cognitive Science* (2016).

[30] Agebson Rocha Façanha, Ticianne Darin, Windson Viana, and Jaime Sánchez. “O&M Indoor Virtual Environments for People Who Are Blind: A Systematic Literature Review”. In: *TACCESS* (2020).

[31] João Guerreiro, Eshed Ohn-Bar, Dragan Ahmetovic, Kris Kitani, and Chieko Asakawa. “How Context and User Behavior Affect Indoor Navigation Assistance for Blind People”. In: *W4A*. 2018.

[32] Lorraine Whitmarsh. “The Benefits of Guide Dog Ownership”. In: *VIR* (2005).

[33] Jillian M Rickly, Nigel Halpern, Marcus Hansen, and John Welsman. “Travelling with a Guide Dog: Experiences of People with Vision Impairment”. In: *Sustainability* (2021).

[34] Pablo-Alejandro Quinones, Tammy Greene, Rayoung Yang, and Mark Newman. “Supporting Visually Impaired Navigation: a Needs-finding Study”. In: *CHI EA*. 2011.

[35] Gaurav Jain, Yuanyang Teng, Dong Heon Cho, Yunhao Xing, Maryam Aziz, and Brian A Smith. “"I Want to Figure Things Out": Supporting Exploration in Navigation for People with Visual Impairments”. In: *PACMHCI* (2023).

[36] Rie Kamikubo, Hernisa Kacorri, and Chieko Asakawa. “" We Are at the Mercy of Others’ Opinion": Supporting Blind People in Recreational Window Shopping with AI-infused Technology”. In: *W4A*. 2024.

[37] João Guerreiro, Dragan Ahmetovic, Daisuke Sato, Kris Kitani, and Chieko Asakawa. “Airport Accessibility and Navigation Assistance for People with Visual Impairments”. In: *CHI*. 2019.

[38] Google. *Google Maps*. Retrieved in July, 2023 from https://maps.google.com. 2023.

[39] Nicholas Giudice, William Whalen, Tim Riehle, Shane Anderson, and Stacy Doore. “Evaluation of an Accessible, Real-Time, and Infrastructure-Free Indoor Navigation System by Users Who Are Blind in the Mall of America”. In: *Journal of Visual Impairment & Blindness* (2019).

[40] Timothy H Riehle, Shane M Anderson, Patrick A Lichter, Nicholas A Giudice, Suneel I Sheikh, Robert J Knuesel, Daniel T Kollmann, and Daniel S Hedin. “Indoor Magnetic Navigation for the Blind”. In: *EMBC*. 2012.

[41] Chris Yoon, Ryan Louie, Jeremy Ryan, MinhKhang Vu, Hyegi Bang, William Derksen, and Paul Ruvolo. “Leveraging Augmented Reality to Create Apps for People with Visual Disabilities: A Case Study in Indoor Navigation”. In: *ASSETS*. 2019.

[42] Sakmongkon Chumkamon, Peranitti Tuvaphanthaphiphat, and Phongsak Keeratiwintakorn. "A Blind Navigation System Using RFID for Indoor Environments". In: *ECTI-CON*. 2008.

[43] Aura Ganz, Siddhesh Rajan Gandhi, Carole Wilson, and Gary Mullett. "INSIGHT: RFID and Bluetooth Enabled Automated Space for the Blind and Visually Impaired". In: *EMBC*. 2010.

[44] Manoj Penmetcha, Arabinda Samantaray, and Byung-Cheol Min. "Smartresponse: Emergency and Non-emergency Response for Smartphone Based Indoor Localization Applications". In: *HCI International – Posters' Extended Abstracts*. 2017.

[45] Madoka Nakajima and Shinichiro Haruyama. "New Indoor Navigation System for Visually Impaired People Using Visible Light Communication". In: *EURASIP Journal on Wireless Communications and Networking* (2013).

[46] Masayuki Murata, Dragan Ahmetovic, Daisuke Sato, Hironobu Takagi, Kris M Kitani, and Chieko Asakawa. "Smartphone-based Indoor Localization for Blind Navigation Across Building Complexes". In: *PerCom*. 2018.

[47] Jee-Eun Kim, Masahiro Bessho, Shinsuke Kobayashi, Noboru Koshizuka, and Ken Sakamura. "Navigating Visually Impaired Travelers in a Large Train Station Using Smartphone and Bluetooth Low Energy". In: *SAC*. 2016.

[48] Hsuan-Eng Chen, Yi-Ying Lin, Chien-Hsing Chen, and I-Fang Wang. "BlindNavi: A Navigation App for the Visually Impaired Smartphone User". In: *CHI EA*. 2015.

[49] Giorgio Presti, Dragan Ahmetovic, Mattia Ducci, Cristian Bernareggi, Luca Ludovico, Adriano Baratè, Federico Avanzini, and Sergio Mascetti. "WatchOut: Obstacle Sonification for People with Visual Impairment or Blindness". In: *ASSETS*. 2019.

[50] Adam J Spiers and Aaron M Dollar. "Outdoor Pedestrian Navigation Assistance with a Shape-changing Haptic Interface and Comparison with a Vibrotactile Device". In: *HAPTICS*. 2016.

[51] Jean-Philippe Choiniere and Clement Gosselin. "Development and Experimental Validation of a Haptic Compass Based on Asymmetric Torque Stimuli". In: *ToH* (2016).

[52] Guanhong Liu, Tianyu Yu, Chun Yu, Haiqing Xu, Shuchang Xu, Ciyuan Yang, Feng Wang, Haipeng Mi, and Yuanchun Shi. "Tactile Compass: Enabling Visually Impaired People to Follow a Path with Continuous Directional Feedback". In: *CHI*. 2021.

[53] David A Ross and Bruce B Blasch. "Wearable Interfaces for Orientation and Wayfinding". In: *ASSETS*. 2000.

[54] Young Hoon Lee and Gerard Medioni. "Wearable RGBD Indoor Navigation System for the Blind". In: *ECCV WS*. 2014.

[55] Shuijing Liu, Aamir Hasan, Kaiwen Hong, Runxuan Wang, Peixin Chang, Zachary Mizrachi, Justin Lin, D Livingston McPherson, Wendy A Rogers, and Katherine Driggs-Campbell. "DRAGON: A Dialogue-based Robot for Assistive Navigation with Visual Language Grounding". In: *RA-L* (2024).

[56] John Morris and James Mueller. "Blind and Deaf Consumer Preferences for Android and IOS Smartphones". In: *Inclusive Designing*. 2014.

[57] Natalina Martiniello, Werner Eisenbarth, Christine Lehane, Aaron Johnson, and Walter Wittich. "Exploring the Use of Smartphones and Tablets Among People with Visual Impairments: Are Mainstream Devices Replacing the Use of Traditional Visual Aids?" In: *Assistive Technology* (2019).

[58] Luyao Wang, Qihe Chen, Yan Zhang, Ziang Li, Tingmin Yan, Fan Wang, Guyue Zhou, and Jiangtao Gong. "Can Quadruped Guide Robots Be Used as Guide Dogs?" In: *IROS*. 2023.

[59] Seita Kayukawa, Tatsuya Ishihara, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. "Guiding Blind Pedestrians in Public Spaces by Understanding Walking Behavior of Nearby Pedestrians". In: *IMWUT* (2020).

[60] Seita Kayukawa, Daisuke Sato, Masayuki Murata, Tatsuya Ishihara, Akihiro Kosugi, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. "How Users, Facility Managers, and Bystanders Perceive and Accept a Navigation Robot for Visually Impaired People in Public Buildings". In: *RO-MAN*. 2022.

[61] Seita Kayukawa, Keita Higuchi, João Guerreiro, Shigeo Morishima, Yoichi Sato, Kris Kitani, and Chieko Asakawa. "BBeep: A Sonic Collision Avoidance System for Blind Travellers and Nearby Pedestrians". In: *CHI*. 2019.

[62] IncNavi. *IncNavi*. Retrieved in September 8, 2024 from https://www.nihonbashi-tokyo.jp/inclusive_navi/. 2024.

[63] Robert K Katzschmann, Brandon Araki, and Daniela Rus. "Safe Local Navigation for Visually Impaired Users with a Time-of-flight and Haptic Feedback Device". In: *TNSRE* (2018).

[64] Hsueh-Cheng Wang, Robert K Katzschmann, Santani Teng, Brandon Araki, Laura Giarré, and Daniela Rus. "Enabling Independent Navigation for Visually Impaired People Through a Wearable Vision-based Feedback System". In: *ICRA*. 2017.

[65] Rabia Jafri, Rodrigo Louzada Campos, Syed Abid Ali, and Hamid R Arabnia. "Visual and Infrared Sensor Data-based Obstacle Detection for the Visually Impaired Using the Google Project Tango Tablet Development Kit and the Unity Engine". In: *IEEE Access* (2017).

[66] Alberto Rodríguez, J Javier Yebes, Pablo F Alcantarilla, Luis M Bergasa, Javier Almazán, and Andrés Cela. "Assisting the Visually Impaired: Obstacle Detection and Warning System by Acoustic Feedback". In: *Sensors* (2012).

[67] N Veeranjaneyulu. "A Meta-Analysis on Obstacle Detection for Visually Impaired People". In: *JPR* (2019).

[68] Angela Constantinescu, Karin Müller, Monica Haurilet, Vanessa Petrausch, and Rainer Stiefelhagen. "Bring the Environment to Life: A Sonification Module for People with Visual Impairments to Improve Situation Awareness". In: *ICMI*. 2020.

[69] Jagannadh Pariti, Vinita Tibdewal, and Tae Oh. "Intelligent Mobility Cane-Lessons Learned From Evaluation of Obstacle Notification System Using a Haptic Approach". In: *CHI EA*. 2020.

[70] Andrés A Díaz-Toro, Sixto E Campaña-Bastidas, and Eduardo F Caicedo-Bravo. "Vision-Based System for Assisting Blind People to Wander Unknown Environments in a Safe Way". In: *Journal of Sensors* (2021).

[71] Carolyn Ton, Abdelmalak Omar, Vitaliy Szedenko, Viet Hung Tran, Alina Aftab, Fabiana Perla, Michael J. Bernstein, and Yi Yang. "LIDAR Assist Spatial Sensing for the Visually Impaired and Performance Analysis". In: *TNSRE* (2018).

[72] Young Hoon Lee and Gérard Medioni. "RGB-D Camera Based Navigation for the Visually Impaired". In: *RSS WS*. 2011.

[73] Vivek Pradeep, Gerard Medioni, and James Weiland. "Robot Vision for the Visually Impaired". In: *CVPRW*. 2010.

[74] José Jesús Guerrero, Ruben Martinez-Cantin, and Carlos Sagüés. "Visual Mapless Navigation Based on Homographies". In: *Journal of Robotic Systems* (2005).

[75] Saeid Nahavandi, Roohallah Alizadehsani, Darius Nahavandi, Shady Mohamed, Navid Mohajer, Mohammad Rokonuzzaman, and Ibrahim Hossain. "A Comprehensive Review on Autonomous Navigation". In: *ACM Computing Surveys* (2025).

[76] Guilherme N DeSouza and Avinash C Kak. "Vision for Mobile Robot Navigation: A Survey". In: *TPAMI* (2002).

[77] Mehmet Serdar Güzel. "Autonomous Vehicle Navigation Using Vision and Mapless Strategies: a Survey". In: *AIME* (2013).

[78] Alexandre Bernardino and José Santos-Victor. "Visual Behaviours for Binocular Tracking". In: *Robotics and Autonomous Systems* (1998).

[79] Philippe Gaussier, Cédric Joulain, Stéphane Zrehen, Jean-Paul Banquet, and Arnaud Revel. "Visual Navigation in an Open Environment Without Map". In: *IROS*. 1997.

[80] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. "Visual Navigation for Mobile Robots: A Survey". In: *Journal of Intelligent and Robotic Systems* (2008).

[81] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. "Target-driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning". In: *ICRA*. 2017.

[82] Ronja Möller, Antonino Furnari, Sebastiano Battiato, Aki Härmä, and Giovanni Maria Farinella. "A Survey on Human-aware Robot Navigation". In: *Robotics and Autonomous Systems* (2021).

[83] Fan Yang, Dung-Han Lee, John Keller, and Sebastian Scherer. "Graph-based Topological Exploration Planning in Large-scale 3d Environments". In: *ICRA*. 2021.

[84] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. "Occupancy Anticipation for Efficient Exploration and Navigation". In: *ECCV*. 2020.

[85] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. "A Survey of Human-in-the-loop for Machine Learning". In: *Future Generation Computer Systems* (2022).

[86] James E Allen, Curry I Guinn, and Eric Horvtz. "Mixed-initiative Interaction". In: *IEEE Intelligent Systems and their Applications* (1999).

[87] Riku Arakawa, Prasoon Patidar, Will Page, Jill Lehman, and Mayank Goel. "Scaling Context-Aware Task Assistants That Learn From Demonstration and Adapt Through Mixed-Initiative Dialogue". In: *UIST*. 2025.

[88] Li Liu, Diji Yang, Sijia Zhong, Kalyana Suma Sree Tholeti, Lei Ding, Yi Zhang, and Leilani Gilpin. "Right This Way: Can VLMs Guide Us to See More to Answer Questions?" In: *NeurIPS* (2024).

[89] Rie Kamikubo, Seita Kayukawa, Yuka Kaniwa, Allan Wang, Hernisa Kacorri, Hironobu Takagi, and Chieko Asakawa. "Beyond Omakase: Designing Shared Control for Navigation Robots with Blind People". In: *CHI*. 2025.

[90] Yue Wang and Fumin Zhang. *Trends in Control and Decision-making for Human-robot Collaboration Systems*. Springer, 2017.

[91] Anh-Tu Nguyen, Jagat Jyoti Rath, Chen Lv, Thierry-Marie Guerra, and Jimmy Lauber. "Human-machine Shared Driving Control for Semi-autonomous Vehicles Using Level of Cooperativeness". In: *Sensors* (2021).

[92] Hiroaki Seki, S Kobayashi, Yoshitsugu Kamiya, Masatoshi Hikizu, and Hisanao Nomura. "Autonomous/semi-autonomous Navigation System of a Wheelchair by Active Ultrasonic Beacons". In: *ICRA*. 2000.

[93] Mahendran Subramanian, Noyan Songur, Darrell Adjei, Pavel Orlov, and A Aldo Faisal. "A. Eye Drive: Gaze-based Semi-autonomous Wheelchair Interface". In: *EMBC*. 2019.

[94] Xavier Perrin, Ricardo Chavarriaga, Francis Colas, Roland Siegwart, and José del R Millán. "Brain-coupled Interaction for Semi-autonomous Navigation of an Assistive Robot". In: *Robotics and Autonomous Systems* (2010).

[95] Gerard Lacey and Shane MacNamara. "Context-aware Shared Control of a Robot Mobility Aid for the Elderly Blind". In: *IJRR* (2000).

[96] Hochul Hwang, Tim Xia, Ibrahima Keita, Ken Suzuki, Joydeep Biswas, Sunghoon I. Lee, and Donghyun Kim. *System Configuration and Navigation of a Guide Dog Robot: Toward Animal Guide Dog-Level Guiding Work*. 2022.

[97] Shinji Kotani, Hideo Mori, and Noriaki Kiyohiro. "Development of the Robotic Travel Aid "HITOMI"". In: *Robotics and Autonomous Systems* (1996).

[98] David Alejo, Gonzalo Mier, Carlos Marques, Fernando Caballero, Luís Merino, and Paulo Alvito. "SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers". In: *Advances in Robotics Research: From Lab to Market* (2019).

[99] Barzin Doroodgar, Yugang Liu, and Goldie Nejat. "A Learning-based Semi-autonomous Controller for Robotic Exploration of Unknown Disaster Scenes While Searching for Victims". In: *IEEE Transactions on Cybernetics* (2014).

[100] A Hong, O Igharoro, Yugang Liu, Farzad Niroui, Goldie Nejat, and Beno Benhabib. "Investigating Human-robot Teams for Learning-based Semi-autonomous Control in Urban Search and Rescue Environments". In: *Journal of Intelligent & Robotic Systems* (2019).

[101] Hongru Tang, Xiaosong Cao, Aiguo Song, Yan Guo, and Jiatong Bao. "Human-robot Collaborative Teleoperation System for Semi-autonomous Reconnaissance Robot". In: *ICMA*. 2009.

[102] Yan Zhang, Ziang Li, Haole Guo, Luyao Wang, Qihe Chen, Wenjie Jiang, Mingming Fan, Guyue Zhou, and Jiangtao Gong. ""I Am the Follower, Also the Boss": Exploring Different Levels of Autonomy and Machine Forms of Guiding Robots for the Visually Impaired". In: *CHI*. 2023.

[103] Anxing Xiao, Wenzhe Tong, Lizhi Yang, Jun Zeng, Zhongyu Li, and Koushil Sreenath. “Robotic Guide Dog: Leading a Human with Leash-guided Hybrid Physical Interaction”. In: *ICRA*. 2021.

[104] Limin Zeng, Björn Einert, Alexander Pitkin, and Gerhard Weber. “Hapticrein: Design and Development of an Interactive Haptic Rein for a Guidance Robot”. In: *ICCHP*. 2018.

[105] Liyang Wang, Jinxin Zhao, and Liangjun Zhang. “NavDog: Robotic Navigation Guide Dog Via Model Predictive Control and Human-robot Modeling”. In: *SAC*. 2021.

[106] Karst MP Hoogsteen, Sarit Szpiro, Gabriel Kreiman, and Eli Peli. “Beyond the Cane: Describing Urban Scenes to Blind People for Mobility Tasks”. In: *TACCESS* (2022).

[107] Karin Müller, Christin Engel, Claudia Loitsch, Rainer Stiefelhagen, and Gerhard Weber. “Traveling More Independently: A Study on the Diverse Needs and Challenges of People with Visual or Mobility Impairments in Unfamiliar Indoor Environments”. In: *TACCESS* (2022).

[108] Nikola Banovic, Rachel L Franz, Khai N Truong, Jennifer Mankoff, and Anind K Dey. “Uncovering Information Needs for Independent Spatial Learning for Users Who Are Visually Impaired”. In: *ASSETS*. 2013.

[109] Michele A Williams, Caroline Galbraith, Shaun K Kane, and Amy Hurst. “"just Let the Cane Hit It" How the Blind and Sighted See Navigation Differently”. In: *ASSETS*. 2014.

[110] Manaswi Saha, Alexander J Fiannaca, Melanie Kneisel, Edward Cutrell, and Meredith Ringel Morris. “Closing the Gap: Designing for the Last-few-meters Wayfinding Problem for People with Visual Impairments”. In: *ASSETS*. 2019.

[111] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. “Vizwiz Grand Challenge: Answering Visual Questions From Blind People”. In: *CVPR*. 2018.

[112] Haobin Tan, Chang Chen, Xinyu Luo, Jiaming Zhang, Constantin Seibold, Kailun Yang, and Rainer Stiefelhagen. “Flying Guide Dog: Walkable Path Discovery for the Visually Impaired Utilizing Drones and Transformer-based Semantic Segmentation”. In: *ROBIO*. 2021.

[113] Qiang Chen, Yinong Chen, Jinhui Zhu, Gennaro De Luca, Mei Zhang, and Ying Guo. “Traffic Light and Moving Object Detection for a Guide-dog Robot”. In: *The Journal of Engineering* (2020).

[114] Alexander Fiannaca, Ilias Apostolopoulous, and Eelke Folmer. “Headlock: a Wearable Navigation Aid That Helps Blind Cane Users Traverse Large Open Spaces”. In: *ASSETS*. 2014.

[115] Mouna Afif, Yahia Said, Edwige Pissaloux, Mohamed Atri, et al. “Recognizing Signs and Doors for Indoor Wayfinding for Blind and Visually Impaired Persons”. In: *ATSIP*. 2020.

[116] Yutaro Yamanaka, Seita Kayukawa, Hironobu Takagi, Yuichi Nagaoka, Yoshimune Hiratsuka, and Satoshi Kurihara. “One-Shot Wayfinding Method for Blind People Via OCR and Arrow Analysis with a 360-degree Smartphone Camera”. In: *MobiQuitous*. 2021.

[117] Aira. *Aira*. Retrieved in July 29, 2024 from https://aira.io/. 2024.

[118] BeMyEyes. *BeMyEyes*. Retrieved in July 29, 2024 from https://www.bemyeyes.com/. 2024.

[119] Rie Kamikubo, Naoya Kato, Keita Higuchi, Ryo Yonetani, and Yoichi Sato. "Support Strategies for Remote Guides in Assisting People with Visual Impairments for Effective Indoor Navigation". In: *CHI*. 2020.

[120] Microsoft. *Seeing AI. A Free App That Narrates the World Around You*. Retrieved in September 9, 2021 from https://www.microsoft.com/en-us/seeing-ai. 2021.

[121] BeMyAI. *Introducing Be My AI*. Retrieved in July 29, 2024 from https://www.bemyeyes.com/blog/introducing-be-my-ai. 2024.

[122] OpenAI. *Hello GPT-4o | OpenAI*. Retrieved in September 8, 2024 from https://openai.com/index/hello-gpt-4o/. 2024.

[123] Ricardo E Gonzalez Penuela, Jazmin Collins, Cynthia Bennett, and Shiri Azenkot. "Investigating Use Cases of AI-Powered Scene Description Applications for Blind and Low Vision People". In: *CHI*. 2024.

[124] Jingyi Xie, Rui Yu, He Zhang, Sooyeon Lee, Syed Masum Billah, and John M Carroll. *Emerging Practices for Large Multimodal Model (LMM) Assistance for People with Visual Impairments: Implications for Design*. 2024.

[125] Ruei-Che Chang, Yuxuan Liu, and Anhong Guo. "WorldScribe: Towards Context-Aware Live Visual Descriptions". In: *UIST*. 2024.

[126] Abigale Stangl, Nitin Verma, Kenneth R Fleischmann, Meredith Ringel Morris, and Danna Gurari. "Going Beyond One-size-fits-all Image Descriptions to Satisfy the Information Wants of People Who Are Blind or Have Low Vision". In: *ASSETS*. 2021.

[127] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. "Vision-and-language Navigation: Interpreting Visually-grounded Navigation Instructions in Real Environments". In: *CVPR*. 2018.

[128] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. "Sim-to-real Transfer for Vision-and-language Navigation". In: *CoRL*. 2021.

[129] Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. "Towards Long-horizon Vision-language Navigation: Platform, Benchmark and Method". In: *CVPR*. 2025.

[130] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. "Reverie: Remote Embodied Visual Referring Expression in Real Indoor Environments". In: *CVPR*. 2020.

[131] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. "Touchdown: Natural Language Navigation and Spatial Reasoning in Visual Street Environments". In: *CVPR*. 2019.

[132] Harm De Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. "Talk the Walk: Navigating New York City Through Grounded Dialogue". In: *arXiv* (2018).

[133] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. "Vision-language Navigation: A Survey and Taxonomy". In: *Neural Computing & Applications* (2023).

[134] Gengze Zhou, Yicong Hong, and Qi Wu. "Navgpt: Explicit Reasoning in Vision-and-language Navigation with Large Language Models". In: *AAAI*. 2024.

[135] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. "Lm-nav: Robotic Navigation with Large Pre-trained Models of Language, Vision, and Action". In: *CoRL*. 2023.

[136] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. "Language Models as Zero-shot Planners: Extracting Actionable Knowledge for Embodied Agents". In: *ICML*. 2022.

[137] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. "Waypoint Models for Instruction-guided Navigation in Continuous Environments". In: *ICCV*. 2021.

[138] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. "NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation". In: *RSS* (2024).

[139] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. "Visual Language Maps for Robot Navigation". In: *ICRA*. 2023.

[140] Harel Biggie, Ajay Narasimha Mopidevi, Dusty Woods, and Christoffer Heckman. "Tell Me Where to Go: A Composable Framework for Context-aware Embodied Robot Navigation". In: *CoRL*. 2023.

[141] Chen-Lung Lu, Zi-Yan Liu, Jui-Te Huang, Ching-I Huang, Bo-Hui Wang, Yi Chen, Nien-Hsin Wu, Hsueh-Cheng Wang, Laura Giarré, and Pei-Yi Kuo. "Assistive Navigation Using Deep Reinforcement Learning Guiding Robot With UWB/Voice Beacons and Semantic Feedbacks for Blind and Visually Impaired People". In: *Frontiers in Robotics and AI* (2021).

[142] Benjamin Poppinga, Charlotte Magnusson, Martin Pielot, and Kirsten Rassmus-Gröhn. "TouchOver Map: Audio-tactile Exploration of Interactive Maps". In: *MobileHCI*. 2011.

[143] William Grussenmeyer, Jesel Garcia, and Fang Jiang. "Feasibility of Using Haptic Directions Through Maps with a Tablet and Smart Watch for People Who Are Blind and Visually Impaired". In: *MobileHCI*. 2016.

[144] Leona Holloway, Kim Marriott, and Matthew Butler. "Accessible Maps for the Blind: Comparing 3D Printed Models with Tactile Graphics". In: *CHI*. 2018.

[145] Apple. *IPhone 12 Pro - Technical Specifications*. Retrieved in January 17, 2022 from https://support.apple.com/kb/SP831?locale=en_US. 2021.

[146] Peter E Hart, Nils J Nilsson, and Bertram Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* (1968).

[147] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018.

[148] Adriano Garcia, Edward Mattison, and Kanad Ghose. "High-speed Vision-based Autonomous Indoor Navigation of a Quadcopter". In: *ICUAS*. 2015.

[149] Adriano Garcia, Sandeep S Mittal, Edward Kiewra, and Kanad Ghose. "A Convolutional Neural Network Vision System Approach to Indoor Autonomous Quadrotor Navigation". In: *ICUAS*. 2019.

[150] Seita Kayukawa, Hironobu Takagi, João Guerreiro, Shigeo Morishima, and Chieko Asakawa. "Smartphone-Based Assistance for Blind People to Stand in Lines". In: *CHI EA*. 2020.

[151] Pannag R Sanketi and James M Coughlan. "Anti-blur Feedback for Visually Impaired Users of Smartphone Cameras". In: *ASSETS*. 2010.

[152] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P Bigham. "Supporting Blind Photography". In: *ASSETS*. 2011.

[153] Seita Kayukawa, Tatsuya Ishihara, Hironobu Takagi, Shigeo Morishima, and Chieko Asakawa. "BlindPilot: A Robotic Local Navigation System That Leads Blind People to a Landmark Object". In: *CHI EA*. 2020.

[154] Nelson Daniel Troncoso Aldas, Sooyeon Lee, Chonghan Lee, Mary Beth Rosson, John M Carroll, and Vijaykrishnan Narayanan. "AIGuide: An Augmented Reality Hand Guidance Application for People with Visual Impairments". In: *ASSETS*. 2020.

[155] Dragan Ahmetovic, Federico Avanzini, Adriano Baratè, Cristian Bernareggi, Gabriele Galimberti, Luca A Ludovico, Sergio Mascetti, and Giorgio Presti. "Sonification of Rotation Instructions to Support Navigation of People with Visual Impairment". In: *PerCom*. 2019.

[156] Microsoft. *SoundScape*. Retrieved in September 9, 2021 from https://www.microsoft.com/en-us/research/product/soundscape/. 2021.

[157] Jacobus C Lock, Iain D Gilchrist, Iain D Gilchrist, Grzegorz Cielniak, and Nicola Bellotto. "Experimental Analysis of a Spatialised Audio Interface for People with Visual Impairments". In: *TACCESS* (2020).

[158] Jack M Loomis, Reginald G Golledge, and Roberta L Klatzky. "Navigation System for the Blind: Auditory Display Modes and Guidance". In: *Presence* (1998).

[159] Jeffrey R Blum, Mathieu Bouchard, and Jeremy R Cooperstock. "What's Around Me? Spatialized Audio Augmented Reality for Blind Users with a Smartphone". In: *MobiQuitous*. 2011.

[160] Seongho Kim, Taeyeon Kim, Choong Sun Kim, Hyeongdo Choi, Yong Jun Kim, Gyu Soup Lee, Ockkyun Oh, and Byung Jin Cho. "Two-Dimensional Thermal Haptic Module Based on a Flexible Thermoelectric Device". In: *Soft Robotics* (2020).

[161] Arshad Nasser, Kai-Ning Keng, and Kening Zhu. "Thermalcane: Exploring Thermotactile Directional Cues on Cane-grip for Non-visual Navigation". In: *ASSETS*. 2020.

[162] Nicholas A Bradley and Mark D Dunlop. "Investigating Context-aware Clues to Assist Navigation for Visually Impaired People". In: *Proceedings of Workshop on Building Bridges: Interdisciplinary Context-Sensitive Computing, University of Glasgow*. 2002.

[163] Shiri Azenkot, Richard E Ladner, and Jacob O Wobbrock. "Smartphone Haptic Feedback for Nonvisual Wayfinding". In: *ASSETS*. 2011.

[164] Manuel Martinez, Angela Constantinescu, Boris Schauerte, Daniel Koester, and Rainer Stiefelhagen. "Cognitive Evaluation of Haptic and Audio Feedback in Short Range Navigation Tasks". In: *ICCHP*. 2014.

[165] Apple Developer. *Displaying a Point Cloud Using Scene Depth*. Retrieved in July, 2023 from https://developer.apple.com/documentation/arkit/environmental_analysis/displaying_a_point_cloud_using_scene_depth. 2021.

[166] Apple Developer. *ARKit*. Retrieved in December 29, 2025 from https://developer.apple.com/augmented-reality/arkit/. 2025.

[167] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. "Real-time Plane Segmentation Using RGB-D Cameras". In: *Robot Soccer World Cup XV*. 2012.

[168] Martin A Fischler and Robert C Bolles. "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Communications of the ACM* (1981).

[169] Jiongtao Xiong, Yijun Liu, Xiangrong Ye, Long Han, Huihuan Qian, and Yangsheng Xu. "A Hybrid Lidar-based Indoor Navigation System Enhanced by Ceiling Visual Codes for Mobile Robots". In: *ROBIO*. 2016.

[170] Masaki Nakamiya, Yasue Kishino, Tsutomu Terada, and Shojiro Nishio. "A Route Planning Method Using Cost Map for Mobile Sensor Nodes". In: *ISWPC*. 2007.

[171] Yi Cheng and Gong Ye Wang. "Mobile Robot Navigation Based on Lidar". In: *CCDC*. 2018.

[172] Bridget A Lewis, Jesse L Eisert, and Carryl L Baldwin. "Effect of Tactile Location, Pulse Duration, and Interpulse Interval on Perceived Urgency". In: *Transportation Research Record* (2014).

[173] Carryl L Baldwin, Jesse L Eisert, Andre Garcia, Bridget Lewis, Stephanie M Pratt, and Christian Gonzalez. "Multimodal Urgency Coding: Auditory, Visual, and Tactile Parameters and Their Impact on Perceived Urgency". In: *Work* (2012).

[174] John Brooke et al. "SUS-A Quick and Dirty Usability Scale". In: *Usability Evaluation in Industry* (1996).

[175] J. Taery Kim, Wenhao Yu, Yash Kothari, Jie Tan, Greg Turk, and Sehoon Ha. "Transforming a Quadruped Into a Guide Robot for the Visually Impaired: Formalizing Wayfinding, Interaction Modeling, and Safety Mechanism". In: *CoRL*. 2023.

[176] Ziwei Xu, Haitian Zheng, Minjian Pang, Yangchun Zhu, Xiongfei Su, Guyue Zhou, and Lu Fang. "Utilizing High-level Visual Feature for Indoor Shopping Mall Navigation". In: *GlobalSIP*. 2017.

[177] Liping Yang and Michael Worboys. "Generation of Navigation Graphs for Indoor Space". In: *IJGIS* (2015).

[178] Simon Schmitt, Larissa Zech, Katinka Wolter, Thomas Willemsen, Harald Sternberg, and Marcel Kyas. "Fast Routing Graph Extraction From Floor Plans". In: *IPIN*. 2017.

[179] Meiqing Fu, Rui Liu, Bing Qi, and Raja R. Issa. "Generating Straight Skeleton-based Navigation Networks with Industry Foundation Classes for Indoor Way-finding". In: *Automation in Construction* (2020).

[180] Yueyong Pang, Liangchen Zhou, Bingxian Lin, Guonian Lv, and Chi Zhang. "Generation of Navigation Networks for Corridor Spaces Based on Indoor Visibility Map". In: *IJGIS* (2020).

[181] Jun Li, Chee Leong Chan, Jian Le Chan, Zhengguo Li, Kong Wah Wan, and Wei Yun Yau. "Cognitive Navigation for Indoor Environment Using Floorplan". In: *IROS*. 2021.

[182] Tomoya Honto, Yoshihiro Sugaya, Tomo Miyazaki, and Shinichiro Omachi. "Analysis of Floor Map Image in Information Board for Indoor Navigation". In: *IPIN*. 2017.

[183] T. Y. Zhang and C. Y. Suen. "A Fast Parallel Algorithm for Thinning Digital Patterns". In: *Communications of the ACM* (1984).

[184] JaidedAI. *JaidedAI/EasyOCR*. Retrieved in July, 2023 from https://github.com/JaidedAI/EasyOCR. 2023.

[185] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. *SpaceNet: A Remote Sensing Dataset and Challenge Series*. 2019.

[186] Edsger W Dijkstra. *A Note on Two Problems in Connexion with Graphs*. Springer-Verlag, 1959.

[187] Apple Developer. *Apple Human Interface Guidelines*. Retrieved in September, 2023 from https://developer.apple.com/design/human-interface-guidelines/buttons. 2023.

[188] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors". In: *CVPR*. 2023.

[189] João Guerreiro, Eshed Ohn-Bar, Dragan Ahmetovic, Kris Kitani, and Chieko Asakawa. "How Context and User Behavior Affect Indoor Navigation Assistance for Blind People". In: *W4A*. 2018.

[190] Sandra G Hart and Lowell E Staveland. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research". In: *Advances in Psychology*. 1988.

[191] Nicholas A. Giudice, Benjamin A. Guenther, Toni M. Kaplan, Shane M. Anderson, Robert J. Knuesel, and Joseph F. Cioffi. "Use of an Indoor Navigation System by Sighted and Blind Travelers: Performance Similarities Across Visual Status and Age". In: *TACCESS* (2020).

[192] Yuanyang Teng, Connor Courtien, David Angel Rios, Yves M Tseng, Jacqueline Gibson, Maryam Aziz, Avery Reyna, Rajan Vaish, and Brian A Smith. "Help Supporters: Exploring the Design Space of Assistive Technologies to Support Face-to-Face Help Between Blind and Sighted Strangers". In: *CHI*. 2024.

[193] Apple. *How to Use AirDrop on Your iPhone or iPad*. Retrieved in July, 2023 from https://support.apple.com/en-us/HT204144. 2023.

[194] Ryan Crabb, Seyed Ali Cheraghi, and James M Coughlan. "A Lightweight Approach to Localization for Blind and Visually Impaired Travelers". In: *Sensors* (2023).

[195] Vladimir Kulyukin, Chaitanya Gharpure, Pradnya Sute, Nathan De Graw, John Nicholson, and S Pavithran. "A Robotic Wayfinding System for the Visually Impaired". In: *AAAI*. 2004.

[196] Nuzhah Gooda Sahib, Tony Stockman, Anastasios Tombros, and Oussama Metatla. "Participatory Design with Blind Users: a Scenario-based Approach". In: *INTERACT*. 2013.

[197] Zhichao Chen and Stanley T Birchfield. "Qualitative Vision-based Path Following". In: *T-RO* (2009).

[198] Piyoosh Mukhija, Siddharth Tourani, and K Madhava Krishna. "Outdoor Intersection Detection for Autonomous Exploration". In: *ITSC*. 2012.

[199] Johan Larsson, Mathias Broxvall, and Alessandro Saffiotti. "Laser Based Intersection Detection for Reactive Navigation in an Underground Mine". In: *IROS*. 2008.

[200] Quanwen Zhu, Long Chen, Qingquan Li, Ming Li, Andreas Nüchter, and Jian Wang. "3d Lidar Point Cloud Based Intersection Recognition for Autonomous Driving". In: *Intelligent Vehicles Symposium*. 2012.

[201] Hiroki Ishida, Kouchi Matsutani, Miho Adachi, Shingo Kobayashi, and Ryusuke Miyamoto. "Intersection Recognition Using Results of Semantic Segmentation for Visual Navigation". In: *ICVS*. 2019.

[202] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. "Text Recognition in the Wild: A Survey". In: *ACM Computing Surveys* (2021).

[203] Mrouj Almuhajri and Ching Y. Suen. "A Complete Framework for Shop Signboards Detection and Classification". In: *ICPR*. 2022.

[204] German Flores and Roberto Manduchi. "Easy Return: an App for Indoor Backtracking Assistance". In: *CHI*. 2018.

[205] Glenn Jocher et al. *ultralytics/yolov5: V6.0 - YOLOv5n 'Nano' Models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support*. Retrieved in August 9, 2025 from https://doi.org/10.5281/zenodo.5563715. 2021.

[206] Kai Zhu and Tao Zhang. "Deep Reinforcement Learning Based Mobile Robot Navigation: A Review". In: *Tsinghua Science and Technology* (2021).

[207] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "Visual Instruction Tuning". In: *NeurIPS* (2023).

[208] Laurel D Riek. "Wizard of Oz Studies in HRI: a Systematic Review and New Reporting Guidelines". In: *Journal of Human-Robot Interaction* (2012).

[209] Saki Asakawa, João Guerreiro, Dragan Ahmetovic, Kris M. Kitani, and Chieko Asakawa. "The Present and Future of Museum Accessibility for People with Visual Impairments". In: *ASSETS*. 2018.

[210] Saki Asakawa, João Guerreiro, Daisuke Sato, Hironobu Takagi, Dragan Ahmetovic, Desi Gonzalez, Kris M. Kitani, and Chieko Asakawa. "An Independent and Interactive Museum Experience for Blind People". In: *W4A*. 2019.

[211] Khadidja Delloul and Slimane Larabi. "Image Captioning State-of-the-Art: Is It Enough for the Guidance of Visually Impaired in an Environment?" In: *CSA*. 2022.

[212] Ouster. *VLP 16*. Retrieved in November 25, 2024 from https://ouster.com/products/hardware/vlp-16. 2024.

[213] Intel. *Intel® RealSense™ Depth Camera D455*. Retrieved in November 25, 2024 from https://www.intelrealsense.com/depth-camera-d455/. 2024.

[214] Intel. *Intel® RealSense™ Depth Camera D435*. Retrieved in November 25, 2024 from https://www.intel.com/content/www/us/en/products/sku/128255/intel-realsense-depth-camera-d435/specifications.html. 2024.

[215] NUC. *Ruby R8 – AMD Ryzen R7-4800U*. Retrieved in November 25, 2024 from https://simplynuc.co.uk/wp-content/uploads/briefs/SimplyNUCProductBrief-CBM1r8RB.pdf. 2024.

[216] Seeed Studio. *Jetson Mate Getting Started*. Retrieved in November 25, 2024 from https://wiki.seeedstudio.com/Jetson-Mate/. 2024.

[217] Cartographer ROS. *Cartographer ROS Integration*. Retrieved in November 25, 2024 from https://google-cartographer-ros.readthedocs.io/en/latest/. 2024.

[218] Pierre Soille. *Morphological Image Analysis: Principles and Applications*. Springer, 1999.

[219] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *KDD*. 1996.

[220] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. "Clip-fields: Weakly Supervised Semantic Fields for Robotic Memory". In: *RSS*. 2023.

[221] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. "VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation". In: *ICRA*. 2024.

[222] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *EMNLP*. 2021.

[223] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *ICML*. 2021.

[224] James C Byers, AC Bittner, and Susan G Hill. "Traditional and Raw Task Load Index (TLX) Correlations: Are Paired Comparisons Necessary". In: *Advances in Industrial Ergonomics and Safety* (1989).

[225] Aaron Bangor, Philip Kortum, and James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *Journal of Usability Studies* (2009).

[226] Sandra G Hart. "NASA-task Load Index (NASA-TLX); 20 Years Later". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2006.

[227] Morten Hertzum. "Reference Values and Subscale Patterns for the Task Load Index (TLX): a Meta-analytic Review". In: *Ergonomics* (2021).

[228] Dingning Liu, Xiaoshui Huang, Yuenan Hou, Zhihui Wang, Zhenfei Yin, Yongshun Gong, Peng Gao, and Wanli Ouyang. *Uni3d-llm: Unifying Point Cloud Perception, Generation and Editing with Large Language Models*. 2024.

[229] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. "Pointllm: Empowering Large Language Models to Understand Point Clouds". In: *ECCV*. 2024.

[230] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik, and Devendra Singh Chaplot. "GOAT: GO to Any Thing". In: *RSS*. 2024.

[231] Allan Wang, Christoforos Mavrogiannis, and Aaron Steinfeld. "Group-based Motion Prediction for Navigation in Crowded Environments". In: *CoRL*. 2022.

[232] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator". In: *CoRL*. 2017.

[233] Dídac Surís, Sachit Menon, and Carl Vondrick. "Vipergpt: Visual Inference Via Python Execution for Reasoning". In: *ICCV*. 2023.

[234] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. "Grounding Dino: Marrying Dino with Grounded Pre-training for Open-set Object Detection". In: *ECCV*. 2024.

[235] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. "Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding". In: *EMNLP*. 2020.

[236] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. "Vision-and-Language Navigation: A Survey of Tasks, Methods, and Future Directions". In: *ACL*. 2022.

[237] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. "Stay on the Path: Instruction Fidelity in Vision-and-language Navigation". In: *ACL*. 2019.

[238] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. "General Evaluation for Instruction Conditioned Navigation Using Dynamic Time Warping". In: *arXiv* (2019).

[239] Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-Tur. "Just Ask: An Interactive Learning Framework for Vision and Language Navigation". In: *AAAI*. 2020.

[240] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. "Video-mme: The First-ever Comprehensive Evaluation Benchmark of Multi-modal Llms in Video Analysis". In: *CVPR*. 2025.

[241] Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. "Iterative Vision-and-language Navigation". In: *CVPR*. 2023.

[242] Takeo Igarashi. "Easy and Fast? Rethinking The Future of Content Creation Tools". In: *UIST Adjunct*. 2025.

[243] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. "Fastvit: A Fast Hybrid Vision Transformer Using Structural Reparameterization". In: *ICCV*. 2023.

Publication and Awards

Journal and Conference Full Papers

Below are the representative works conducted by Masaki Kuribayashi during his enrollment at Waseda University. Please refer to his website for other papers[1]

1. Masaki Kuribayashi*, Seita Kayukawa*, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima (* - equal contribution). 2021. LineChaser: A Smartphone-Based Navigation System for Blind People to Stand in Line. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI 2021)

2. Masaki Kuribayashi, Seita Kayukawa, Jayakorn Vongkulbhisal, Daisuke Sato, Chieko Asakawa, Hironobu Takagi, and Shigeo Morishima. 2022. CorridorWalker: Mobile Indoor Walking Assistance for Blind People to Avoid Obstacles and Recognize Intersections. In Proceedings of the 24th International Conference on Human-Computer Interaction with Mobile Devices and Services. (Mobile HCI 2022)

3. Masaki Kuribayashi, Tatsuya Ishihara, Daisuke Sato, Jayakorn Vongkulbhisal, Karnik Ram, Seita Kayukawa, Hironobu Takagi, and Shigeo Morishima, and Chieko Asakawa. 2023. PathFinder: Designing a Map-less Navigation System for Blind People in Unfamiliar Buildings. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI 2023)

4. Yuka Kaniwa*, Masaki Kuribayashi*, Seita Kayukawa, Daisuke Sato, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima (* - equal contribution). 2024. ChitChatGuide: Conversational Interaction Using Large Language Models for Assisting People with Visual Impairments to Explore a Shopping Mall. In Proceedings of the 26th International Conference on Human-Computer Interaction with Mobile Devices and Services. (Mobile HCI 2024)

5. Masaya Kubota*, Masaki Kuribayashi*, Seita Kayukawa, Hironobu Takagi, Chieko Asakawa, and Shigeo Morishima (* - equal contribution). 2024. Snap&Nav: Smartphone-based Indoor Navigation System For Blind People via Floor Map Analysis and Intersection Detection. In Proceedings of the 26th International Conference on Human-Computer Interaction with Mobile Devices and Services. (Mobile HCI 2024)

6. Masaki Kuribayashi, Kohei Uehara, Allan Wang, Shigeo Morishima, and Chieko Asakawa. 2025. WanderGuide: Indoor Map-less Robotic Guide for Exploration by Blind People, In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI 2025)

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

7. Masaki Kuribayashi*, Kohei Uehara*, Allan Wang, Daisuke Sato, Renato Ribeiro, Simon Chu, and Shigeo Morishima. (* - equal contribution) 2025. Memory-Maze: Scenario Driven Visual Language Navigation Benchmark for Guiding Blind People. IEEE Robotics and Automation Letters (RA-L)

Awards

1. WISS Best Paper Award, Workshop of Interactive Systems and Software (WISS), 2020
2. Azusa Ono Memorial Award, Waseda University, 2020
3. Azusa Ono Memorial Award, Waseda University, 2024