Department of Electrical Engineering and Information Technology
Institute for Media Technology
Distributed Multimodal Information Processing Group
Prof. Dr.-Ing. Eckehard Steinbach

Towards Improving Instruction Presentation for Indoor Navigation

Optimierung der Anweisungsdarstellung für Indoor-Navigation

Simon Mang

Master Thesis

Author:	Simon Mang
Address:	[redacted]
Matriculation Number:	[redacted]
Professor:	Prof. Dr.-Ing. Eckehard Steinbach
Advisor:	Prof. Dr. Matthias Kranz
	Dipl.-Medieninf. (Univ.) Andreas Möller

Begin:	18.10.2012
End:	18.04.2013

Department of Electrical Engineering and Information Technology
Institute for Media Technology
Distributed Multimodal Information Processing Group
Prof. Dr.-Ing. Eckehard Steinbach

Declaration

I declare under penalty of perjury that I wrote this Master Thesis entitled

Towards Improving Instruction Presentation for Indoor Navigation
Optimierung der Anweisungsdarstellung für Indoor-Navigation

by myself and that I used no other than the specified sources and tools.

Munich, April 17, 2013

Simon Mang

Simon Mang
[redacted]

Kurzfassung

In den letzten Jahren gewannen Indoor-Navigation Systeme immer mehr an Popularität. Diese Arbeit zielt darauf ab, die Darstellung von Anweisungen im Kontext von Indoor-Navigation zu verbessern. Es wird evaluiert wie und wo Anweisungen angezeigt werden müssen, um das Finden des richtigen Weges zu vereinfachen und gleichzeitig alle Nutzerwünsche zu erfüllen. Eine bestehende Nutzeroberfläche, die diese Bedingungen erfüllt, wird übernommen und angepasst.

Diese Arbeit präsentiert verschiedene Konzepte zur Anweisungsdarstellung. Die Konzepte beinhalten einen Vergleich von Pfeil- und Karten-basierten Darstellungen und verschiedene Modi, die die nötige Häufigkeit von neuen Anweisungen analysieren. Drei Modi werden realisiert um Nutzerpräferenzen hinsichtlich automatisch oder manuell gewechselten Anweisungen zu evaluieren. Während zweier Nutzerstudien werden diese Modi iterativ verbessert.

Es wird gezeigt, dass automatisch gewechselte Anweisungen Nutzer schneller zum Ziel führen als manuell gewechselte. Die Ergebnisse beinhalten außerdem, dass Anweisungen an Entscheidungspunkten ausreichend sind vor allem wenn die Navigation durch eine Kartendarstellung unterstützt wird. Des Weiteren wird aufgezeigt, dass eine Karte allein nicht ausreicht, sie jedoch in Kombination mit Panoramadarstellungen von Bildern einzelner Wegpunkte einen vielversprechenden Ansatz zur Indoor-Navigation darstellt.

Abstract

In recent years pedestrian indoor navigation gained more and more popularity. This thesis aims at improving the instruction presentation. It is evaluated how and where way instructions have to be presented in order to simplify the wayfinding task while satisfying the user's needs at the same time. An existing user interface meeting this requirements is adopted and further extended.
Concepts for different instruction presentations and specialized instruction modes are presented. The presentational concepts include a comparison of arrow or map-based instructions and the different modes help analyzing the necessary frequency of instruction updates. Three modes are realized in order to evaluate users preferences towards automatically or manually updated instructions. In the course of two user studies these mode are iteratively improved.
It is found that automatically updating instructions guides users faster to the goal than doing the same manually. Further, presenting new instructions at decision-points only is sufficient especially if the navigation process is supported by a map presentation. The map alone is not found to be sufficient for an indoor navigation task but in combination with panoramic renderings of images at particular route locations it represents a promising approach for efficient indoor navigation.

Contents

Chapter 1.

Introduction

Recent years have seen an increasing popularity of mobile devices and smart phones supporting navigational systems. Widespread car navigation systems do not simply guide drivers to their destination, they provide radio stations with reliable data to improve public traffic information. Nowadays, navigational help is not restricted to specialized mobile devices anymore. The stores of mobile OS providers are full of navigational applications. Software is no longer limited to provide only car navigation. Pedestrian navigation enjoys great popularity, too. It is therefore not surprising that leading trend research[1] sees indoor navigation as one of the upcoming trends of this decade.

Outdoor navigation systems are usually based on GPS localization. Unfortunately, its accuracy is not sufficient to track people inside buildings. A variety of indoor positioning solutions has been developed. For example, there are systems relying on radio-based [8–10], visual-marker [2] or infrared [7] technologies. However, this thesis incorporates a vision-based localization approach. This technique matches images captured by the user to reference images with known spatial data and locates the user at the position of the best matching reference image. Regarding the growing computational power of mobile phones and their increased efficiency in terms of high-resolution cameras and battery life makes it a promising approach. One striking advantage of the vision-based technology is that costly infrastructure like sensors or terminals becomes superfluous as soon as the reference database is recorded.

The motivation of this thesis is to investigate the human-computer interaction with an existing vision-based indoor navigation system. It is to be explored how different instruction presentations can improve user satisfaction while simplifying the wayfinding task. In the scope of this thesis particular direction instructions are presented in form of symbols (mainly arrows). In contrary to text instructions, which can be difficult to read on small screens, or voice instructions, which might be overheard in noisy environments, symbols are easy and fast to perceive. This thesis evaluates different kinds of instruction presentation and the frequency of instruction updates. How do users rate the presentation of instructions in a VR view compared to a map presentation?

[1] http://www.trendone.com/en/trends/mega-trends.html, last accessed: 04/16/2013

Do they need instructions with each step or are instructions at turns and other possibilities to choose another path sufficient? Is manually changing instructions favored (reactive) or should automatically updated instructions (proactive) be provided?

This thesis' contribution to the questions outlined above is to develop concepts which serve as sources of information for eventual responses and to realize them in actual implementations. The concepts are iteratively improved in the course of two user studies. Findings of a initial study are considered and integrated and the improvements are evaluated again in a follow-up study. The gathered insights are presented, interpreted and resulting suggestions in terms of further improvements and enhancements are highlighted.

After this introduction follows an overview of existing work related thematically to this thesis. Localization techniques as well as considerations about user interfaces in pedestrian navigation systems are outlined. In Chapter 3 the former highlighted concepts are formulated explicitly and the different instruction presentations and modes are explained in detail. An overview of the extended application and the implementation of new and necessary system elements is given in Chapter 4. The realizations are subdivided into three main parts. Section 4.2 illustrates the implementation of the instruction modes, Section 4.3 documents code necessary for the map presentation and in Section 4.4 other important system elements are highlighted.

The evaluations included in this work are treated in Chapters 5 and 6. For each study first the research questions are listed and then the study setup is explained. After a summary of participant data a detailed analysis for all research questions follows. Each result section is finally concluded by a short summary. A discussion of the accumulated findings exists for each study. Chapter 7 compares the insights of both studies, interprets the findings and suggests possible improvements. The thesis ends in a general conclusion of all gathered results and finally gives an outlook of future work in the field of indoor navigation.

Chapter 2.

State of the Art

The following section presents an overview of existing work related to this thesis. First a summary of work dealing with the problem of localization and positioning in a pedestrian context is provided. Afterwards sensor- and vision-based localization techniques are surveyed. Section 2.2 is concerned with User Interface (UI) considerations. After reflecting general UI findings, especially landmark- and map-based UIs are focused. At last there is an overview of related work not fitting the former categories.

2.1. Pedestrian Navigation and Positioning Techniques

In general there are two main fields of pedestrian navigation. On the one hand there are systems helping users to find their destination outside in streets, parks or large areas like an university campus [1]. On the other hand pedestrian navigation is used inside buildings [2]. Intermediate systems working outdoors as well as indoors exist, too [3]. While the Global Positioning System (GPS) is sufficient and well tested for outdoor positioning tasks it lacks precision in indoor settings. In addition to its general inaccuracy of about 15 meters[1] one would have to deal with attenuation effects and multi-path propagation of the signal [4]. Some older work comparing multiple indoor positioning possibilities like Zeimpekis [5] relied on special devices based on a wireless infrastructure. Newer work integrates taxonomies of e.g. light, vision or acoustic sensors. Such a comparative classification benefiting from technical progress and accumulated insights and knowledge is presented by Werner [6]. In his work about ubiquitous navigation he provides a detailed summary of positioning methods and fundamentals for indoor navigation.

2.1.1. Sensor-based Positioning

As GPS is not available indoors, many alternatives have been provided. There are systems relying on infrared (IR) sensors [7], radio frequency networks [8, 9], wireless technology like WLAN or

[1] http://www.kowoma.de/en/gps/accuracy.htm (last accessed: 03/26/2013)

DECT [10] and yet others that for example use an Inertial Measurement Unit (IMU) or ultrasonic signals [11]. The works of Hightower and Borriello [12] and Liu et al. [13] give a survey and taxonomy of localization systems and both analyze properties like accuracy and precision or costs of those systems. A comparison and analysis of mobile indoor navigation systems based on an evaluation framework is given by Huang and Gartner [14].

2.1.2. Vision-based Localization

The technologies mentioned in section 2.1.1 always need a base infrastructure. Either WLAN access points or some kind of markers (optical or radio-frequency based) have to be installed and maintained. In contrary to sensor-based positioning vision-based localization as explained in DeSouza and Kak's work on indoor navigation for robots [15] uses reference images to locate users and is therefore independent from further infrastructure once the reference database is recorded. According to [15] the localization process in vision-based systems consists of four steps. First sensory information has to be acquired, i.e. digital images have to be recorded. Second, the recorded images are analyzed and significant regions are detected. In a third step these observations have to be compared to the database. Finally, in the fourth step, the spatial information of an eventual matching image is mapped to the recorded image from step one. Step two and three can be solved by image feature detection algorithms like SIFT [16] or SURF [17].

The vision-based systems can be subdivided into Augmented Reality (AR) and Virtual Reality (VR) applications. AR systems, like the name suggests, augment and enrich the reality with features. Live captures of hallways and rooms are overlaid with floorplan images in a system provided by Hile and Borriello [18]. In contrary, VR systems are based on pre-recorded images. Liu et al. [19] developed virtual reality environments not based on realtime images but on static 360° panorama captures. This Panorama-Based Virtual Environment (PBVE) may then be explored from a distant desktop computer. Other applications using panorama images are presented by Werner et al. [4] who use SURF for feature extraction and by Kawaji et al. [20]. In their work Kawaji et al. compared SIFT, PCA-SIFT and SURF algorithms with respect to processing time and precision rate. As modern mobile phones contain multicore processors and gain remarkable performance mobile usage scenarios of these algorithms become possible [21]. For evaluation purposes of those PBVEs Schellenbach et al. [22] provide an evaluation framework in a virtual environment based on a treadmill and a large screen.

2.2. User Interaces (UI)s for Pedestrian Navigation

According to Wahlster et al. [23] the design of pedestrian navigation systems significantly differs from car navigation systems. The latter work exclusively in a limited user context with a fixed user position (sits at the steering wheel) and a focused user attention area (looks concentrated onto the street). Contrarily the user context for pedestrian systems varies a lot more. Wahlster et al. mention that route descriptions for business people heading as fast as possible to the train station differ significantly from descriptions for tourists for whom the station is only one out of many destinations. Thus it is important to consider existing interface solutions with a pedestrian context. In this section related work providing indoor and outdoor systems for pedestrians is presented.

2.2.1. UI Considerations for Indoor and Outdoor Navigation

A hybrid system, called BMW Personal Navigator (BPN), combining pedestrian and in-car usage is presented by Krüger et al. [24]. The BPN project, a BMW cooperation project, supplies specialized UIs for desktop, in-car and on-foot use. The different functionalities are not integrated in one unique interface but are spread into several views which are specialized in distinct tasks. The displayed information reaches from weather forecast (pedestrian), over parking slot reservation (car) to route planning (desktop). In the mobile context the user may additionally switch between different view presentations. A top-down view of the scene, an isometric view and an egocentric view are available.

The REAL system [23] was designed to serve the combination of different means of transportation. The result was a system adapting to the changing user situation and taking into account the technical properties of the output device. Two sub-systems are incorporated: an indoor and an outdoor component. The IRREAL sub-system is based on IR beacons and runs on a PDA carried by the user. Outside buildings the ARREAL part is used. This augmented reality AR device consists mainly of a head-mounted display and backpack with a subnotebook and a GPS receiver. Both systems adapt the presentation of route instructions to the technical resources of the output device.

Three UI management techniques important for creating effective Mobile Augmented Reality System (MARS) are presented by Höllerer et al. [25]. *Information filtering* is the task preventing visual overload. Again user context adaption plays an important role. Also the localization accuracy is taken into account and the AR view is adjusted accordingly. If the position tracking is precise hallways and office rooms are virtually labeled. If it is not, a model of the building in miniature is shown. This is what Höllerer et al. name *UI component design*. Simple labeling approaches for objects in AR settings produce overlapping labels and labels for objects not currently visible to users. *View management* prevents annotations to objects outside the viewport and assures unambiguity. As the authors state the *"benefits of MARS will only be achieved if the user interface (UI) is actively managed so as to maximize the relevance and minimize the confusion of the virtual material relative to the real world"* [25].

Möller et al. [26] introduce an indoor navigation system incorporating AR and VR components. The AR interface is evaluated to be more accurate whereas users rate the VR component to be more reliable. The authors argue that for vision-based navigation a combination of both systems is beneficial. As vision-based navigation relies on high quality of the recorded images Möller et al. included interface elements which motivate users to hold the mobile device upright and thus ensure that sufficient visual features can be detected by the system.

Implications for the design of mobile navigation UIs are presented in the work of May et al. on pedestrian navigation aids [27]. It is evaluated what kind of route information is relevant to the user. Especially at key decision-points the interface should display instructions with a certain relation to real world objects (e.g. "turn left at the bank"). Highlighting landmarks, like a salient building, with labels is more popular than giving users information about distance or road names. However, Möller et al. [26] found that if the UI shows information about the remaining distance, time or number of turn to the goal users prefer seeing the remaining distance.

2.2.2. Landmark-based Navigation

As stated in the former paragraph the work of May et al. [27] indicates the importance of landmarks. The following section deals with characteristics of "good" landmarks and how they are used for navigational purposes. In general a landmark may be defined as a notable geographical object that marks a locality and therefore serves as a reference point [28]. Different landmark definitions can be found in a variety of existing work [28–31].

Lovelace et al. [30] analyzed the quality of route descriptions based on a study collecting data about how people give route directions. The number of route segments, turns and landmarks was identified as a sign of quality. Lovelace et al. figured out four landmark classes. *Choice point* landmarks are located at decision-points where maneuvers (e.g. a left turn) are necessary. In contrary *potential choice point* landmarks can be found at decision-points with possibilities to choose an alternative path where the actual route is not left. Landmarks along the route between turns are summarized as *on route* landmarks whereas *off route* landmarks are often located far away from the path.

Another work characterizing landmarks with respect to real and electronic spaces was published by Sorrows and Hirtle [31]. As there are several characteristics of landmarks like singularity, prominence, accessibility or content the authors stress the importance of a landmark topology. Three types of landmarks are identified: visual, cognitive and structural landmarks. *Visual* landmarks are landmarks with unique visual characteristics. Historical or cultural important landmarks or objects with a different content than visually similar objects are called *cognitive landmarks*. A *structural* landmark is a prominently placed hyperlink in electronic space or a structural important place in spatial space (Trafalgar Square, London).

Motivated by the usage of Location Based Services (LBS) on PDA devices and mobile phones Elias [32, 33] worked on an extension for route directions with landmarks. Elias states that the selection of objects used as landmarks in path descriptions have to be adapted to the type of movement of users (in-car, on-foot). Additionally she provides different techniques for highlighting landmarks. The coloring, resizing and aggregating concepts put focus to landmarks according to the pre-attentive vision processing of humans [34].

In the context of automated route description the computational extraction of landmarks is interesting. Brunner and Radoczky [35] or Lovelace et al. [30] give an overview of algorithm based landmark derivation. Brunner and Radoczky [35] calculate prominent decision-points in a university building and discuss the concept of active landmarks. Active landmarks are usually not visible to the user but communicate via wireless technology with a mobile device. Heuristics and an incremental algorithm to select the best image of a landmark for any location on the path are provided by Hile et al. [1, 36]. Also Raubal and Winter [29] present a method to extract landmarks from a database automatically and how to integrate the extracted landmarks in wayfinding instructions. It is claimed that navigational systems enhanced by landmarks are closer to concepts of wayfinding of human users, more adaptive for individual users and more flexible for different tasks.

2.2.3. Map-based Navigation

According to Kray et al. [37] presenting route information on 2D maps is the most common way of giving wayfinding instructions. Indeed, quite everybody has at least once used a city map and the typical, perspective 2D map view from car navigation systems is nowadays not only known by car drivers. In the next section the use of maps for pedestrian navigation will be described.

Münzer [38] examined the spatial knowledge resulting from using a supportive navigation assistance system in relation to the spatial knowledge users of a map gained. Four conditions were tested: visual+context, auditory+context, auditory, map. The two contextual conditions showed map segments on a PDA with visual (e.g. an arrow pointing left) or auditory (e.g. "turn left") cues. The auditory condition provided the user with auditory hints only. Users in the map condition received a map of the route. After the wayfinding task the users were tested on their route memory performance (correctly remembered directions) and survey knowledge (correctly located thumbnail pictures). Map users not only acquired better survey knowledge, they were even found to have better route knowledge.

The efficiency of different levels of map abstraction was evaluated in a paper on "Schematic vs. topographic maps in pedestrian navigation" [39]. In this study by Gartner and Radoczky participants were presented different kinds of maps (schematic and topographic) that contained different levels of information. Beside the task of walking to the given destination users had to redraw the presented map. This led to the insight that locals provided maps pointing northwards with more information while foreigners' maps lacked information and pointed to the initial walking direction. This indicates that different kinds of maps are better suited for distinct situations as well as that the presentation form influences the generation of mental maps of the environment.

The impacts on UI design including a map overview were analyzed by Hornbæk et al. [40]. They examined navigation patterns and usability consideration in Zoomable User Interface (ZUI) with and without an overview. In terms of accuracy, task completion time and satisfaction ZUIs providing an overview were rated better. As 80% prefer ZUIs including a map overview the authors recommend to consider the provision of overview areas. Furthermore these areas should provide zoom and pan functionality, should be coupled to the detail view and should be at least one-sixteenth the size of the detail window.

In the context of map views the afore-mentioned works of Baus et al. [3] and Krüger et al. [24] have to be quoted again. Baus et al. provide the user not only with a system adapting to changing precision of positional and orientational information, they offer a possibility to change the map displaying the users location, too. Dependent on the walking speed of the user the viewport of the map is zoomed and the number of labels is reduced or incremented. An adaption to the user context is also provided by the BPN [24] system. According to their preferences users may choose between three different map projections: a 2D map view, a perspective view with 3D landmarks and an egocentric 3D view.

Following Butz et al. [7] successful way descriptions help users with knowledge about their position and orientation to re-orientate again. They offer an approach for presenting map instructions in an indoor setting and observe that during the reorientation process users turn the way directions until their egocentric view matches the allocentric view of the map. Another point to mention is the way how the provided system copies with restrictions in the mobile context. The indoor maps and floorplans are designed using vector graphics to save memory and bandwidth. Rendering the graphics is done incrementally, so that important parts (e.g. arrows depicting the path) are rendered first and gradually further parts (walls, rooms, furniture, etc.) are included.

2.3. Other Related Work

This section lists work related to this thesis which cannot be assigned to the above categories. In section 2.2.3 it became clear that map-based navigation is a common way of providing navigation instructions and that users are familiar with this type of interface. A map displaying the whole route is always a kind of *path preview* giving an overview. The concept of previewing the future route was utilized in the Bum Bag Navigator system [41]. At any point of the route users have the possibility to activate a virtual flight to the destination. This camera flight shows the entire path to come and gives additional confidence to the user. Mulloni et al. [2] present a sophisticated interface with different views (World In Miniature (WIM), 2D map and AR). The navigation system is enhanced with markers glued to the floor (called info points). As the camera of the mobile device captures the marker the system knows where the user is and updates the view. The view is divided into two parts. The upper part displays the map view (or AR or WIM). In the lower section users see a preview of future maneuvers. The next action (turn left, go ahead, one stair down, etc.) as well as the number of steps between two actions and the already passed actions are displayed. Beside the fact that the interface was evaluated as an effective one, it was worked out that adding info points to the route increased performance in the means of steps per path and navigation errors. The path preview reminds of the prominent driving directions feature of Google Maps which shows a step-by-step description of the selected route[2].

Another conclusion and design implication from other pedestrian navigation system evaluations is that *position and location confirmation* is useful to users [27]. May et al. also state that users want a confirmation after having made a correct decision. Bhasker et al. [42] showed that the process of location confirmation may also be initiated by users to help the system to correct wrong localizations. After noticing incorrectly displayed locations users may click on the correct location and future localizations are then derived from the corrected location. That users do not want to search their own location by hand but have the desire that a system displays their current location is the reason why map-based UIs provide a possibility to center the map at the user's location. The "My Location" feature of Google Maps[3] or the map component of Cyberguide [43] are such examples.

A framework offering a virtual evaluation environment supporting mobile application development is presented by Diewald et al. [44]. The provided software offers the possibility to evaluate mobile applications in a 3D simulator with connected sensor emulation. This facilitates testing and creating of mobile applications. Using this framework, in theory, the development of indoor navigation systems can be accelerated as long lasting field studies can be skipped. Möller et al. [45] present another tool simplifying the evaluation of indoor navigation applications by replacing the complex localization logic by a *wizard-of-oz* [46] setting. The special evaluation design allows testing at an early stage of development and thus provides early user feedback.

[2] http://support.google.com/maps/bin/answer.py?hl=en&answer=1727367&topic=1687356&ctx=topic (last accessed: 03/28/2013)
[3] http://support.google.com/maps/bin/answer.py?hl=en&answer=153802 (last accessed: 03/28/2013)

Chapter 3.

Concept

This chapter presents the conceptual considerations on instruction presentation for indoor navigation. Improvements and approaches are explained and their advantages are motivated.

3.1. Vision-based Indoor Navigation

The background for this thesis' concepts is a vision-based localization approach. Based on a database of reference images and live captures from the integrated camera the position of the user's mobile phone is determined. In general, the localization process requires four steps [15]:

1. **Image capture**
   At the user's current position the camera records an image of the actual environment
2. **Feature detection and extraction** After the captured image has been transfered to the server image features are detected and extracted using interest point detectors [16, 17, 21]
3. **Feature matching** The calculated image feature descriptor is compared to the entries in the reference database. The best match is selected.
4. **Localization** As spatial information for the entries in the reference database is available the location of the selected best match from step 3 is returned to the user.

Figure 3.1 illustrates this process schematically. The reference images used in this thesis were recorded during the TUMindoor project [48] with a custom-built trolley based on a Ladybug3[1] camera setting (see Figure 4.1a). The campus-wide dataset[2] covers 5236 locations which are 0.5 to 1 meter apart and spread over a track length of 4522 meters.

Once a reference database as highlighted above exists no additional infrastructure is needed. This distinguishes vision-based systems from other positioning techniques which require the adminis-

[1] http://www.ptgrey.com/products/ladybug3/ladybug3_360_video_camera.asp
(last accessed: 03/29/2013)
[2] http://navvis.de/dataset/ (last accessed: 03/31/2013)

Figure 3.1.: Vision-based localization concept.

tration and maintenance of sensors and markers. Furthermore, vision-based localization resembles the human perception which is mainly based on visual information, too. Hence, those systems can be understood rather easily by users and e.g. concepts forcing them to hold the device in a specific position [26] are better accepted.

3.2. Instruction Presentation

The goal of this thesis is to improve instruction presentation for indoor navigation systems. An instruction in this context is the recommendation to take a particular way. This work uses arrows to indicate the correct walking direction. Other concepts like text or voice messages are left for future applications. The present prototype incorporates two different kinds of instruction presentation. One shows an arrow in a VR scene. The other displays arrow markers on a map.

The VR scene is a panoramic rendering of the images available at each location of the above mentioned dataset. This work refers to such renderings as *panoramas*. Each panorama can be rotated horizontally so that users can change their virtual viewport to all directions. Panoramas are overlaid by a red arrow representing the way instruction at the current location. The arrow points either right, left or straight ahead. In addition, each panorama displays the distance to the next turn and the goal. The distance to the next turn is measured in meters which helps anticipating the next maneuver. A progress bar displays the distance to the goal providing users with generalized information about their position on the path (Figure 3.2a).

In section 2.2.3 it becomes clear that providing users with map views in navigational tasks bears advantages. Maps are well-known helper tools for orientating and providing an overview of the route, the environment and the relation between both. Thus maps lead to better knowledge of the path and its environment [38]. In other words, maps simplify wayfinding tasks and enhance performance in the meaning of time and errors made when navigating from A to B.

A map view of the route seems an adequate tool to put users and their position into relation with the building they are in. As they can see their own position, the next turns, the future route, the destination and the building structure and environment enough reference points for a straight and secure navigation are laid out. To take advantage of the visual information provided by panoramas and the spatial information contained in maps both kinds of instruction presentation are displayed at the same time. By default the map interface is minimized and placed in a corner of the screen (Figure 3.2b). In order to give users the possibility to exploit the powers of a virtual map, basic interaction techniques have to be provided. Users have to be able to rotate the map, to maximize and minimize the thumbnail map preview and to pan and zoom the map viewport.

When designing the interaction technique for rotating the map view, two possibilities become obvious. Either users rotate the map by interacting on the map itself with some kind of gesture or they rotate the device and the map is adjusted accordingly. In the background of former works the latter interaction technique is chosen. A self rotating map preserves users' cognitive resources because the map is always pointing into their walking direction [37]. Moreover the process of reorientation is made faster as way descriptions (e.g. a turn arrow) can be interpreted more easily [7]. For people who do not know an environment (indoor or outdoor) it seems to be convenient to rotate their mental model of the route until it matches their walking direction [39].

The second interaction concept concerning the map view is the possibility to enlarge the map area which is by default displayed in a corner of the screen. A maximized map shows more route details and provides more reference-points in the environment. The maximized map covers the entire panorama view. To prevent irritation resulting from an immediate view change and to provide an overall nice look and feel in the application the resizing of the map is animated.

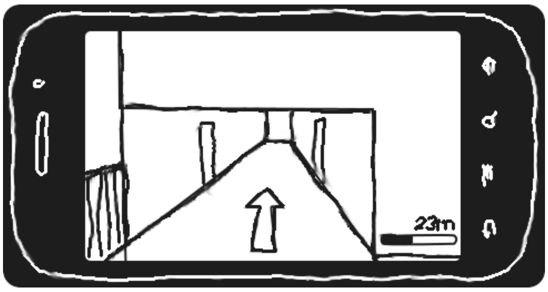
Third, users should be able to pan and zoom the map viewport. This enables them to preview parts of the route clipped by the view borders. Zooming additionally lets users get a bigger picture of the relationship between their current position, the path and the building structure. As highlighted earlier this extra knowledge is assumed to improve performance in wayfinding tasks.

Another concept behind providing route information in a map is the question what kind of presentation the users will prefer. Does the familiarity of a map outperform the visual navigation hints of the 360° panoramas? Do users rather apply the map because they can preview the future path without losing their current location? Or are they relying on the VR instructions with the implication that a continuous mapping between virtual world and real environment is essential?

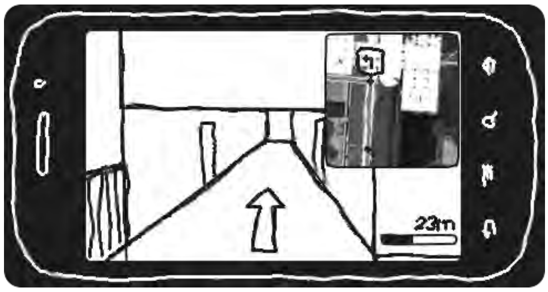
It has to be noted here that the map concept is applied to the instruction modes in the follow-up study only. The instruction modes in initial study are based on the arrow, distance and panorama presentations.

3.3. Instruction Modes

In the scope of this thesis an instruction mode defines when and how often instructions are updated. In order to decide in which situations and in what frequency way instructions have



(a) Sketch of a VR scene with arrow instruction.



(b) Sketch of the same panorama with map presentation and arrow marker.

Figure 3.2.: Sketches of a VR scene with different instruction presentations.

to be updated for navigating efficiently indoors, three different concepts are evaluated in this work. But before these modes are explained in detail the concept of a decision-point has to be introduced. A decision-point is any possibility on a path or route where one can choose between different alternatives how to continue. This can be hallway crossings, forks leading away from a corridor, halls with several exits and so on. Decision-points are found to be essential for human way description [27]. Hence, they play a key role in the path definitions and instruction modes of this thesis.

3.3.1. Fully Automatic Mode

The fully automatic mode displays route instructions continuously. The above mentioned dataset is filtered so that users are provided with new information about every three meters. This does not necessarily result in an update of the entire instruction presentation. While walking down a hallway the direction arrow pointing straight ahead is not changed but the information about the distance to the next decision-point and the panorama in the VR scene is updated continuously. When approaching a decision-point the arrow begins to bend to indicate the proximity of a maneuver and finally points to the new walking direction. The user interaction is mainly to turn the panorama view and look around in the virtual environment. All information is automatically provided by the system. Users do not have to do anything else in this mode than walking along the displayed route according to the instructions provided by the system.

3.3.2. Decision-Point Mode

Like the name suggests the decision-point mode displays panoramas and instructions of decision-points only. No intermediate locations between two decision-points are displayed. Equally to the fully automatic mode users receive all information automatically without interaction. As soon as they have passed a decision-point the panorama and the instruction of the following decision-point is rendered. The idea is to reduce the cognitive load resulting from the constantly updated panoramas and instructions in the fully automatic mode at the costs of a less detailed route description. While the fully automatic mode shows updated information at each path location, including locations of a hallway without any turns, the decision-point mode displays only the locations at the both ends of the hallway.

As both automatic modes (the fully automatic and the decision-point mode) update the instruction presentation without interaction by users, those have to be notified about instruction updates in case they are not observing the device. Hence, at least at decision-points the mobile phone gives tactile feedback in form of rhythmic vibrations in order to notify users about changed way instructions.

3.3.3. Manual Mode

The manual mode allows users to select panoramas manually. The goal is to give users the possibility to switch panoramas on their own. The assumption is that changing panoramas and instructions manually at any time is more convenient than being forced to react on automatic updates. The users are able to preview the path based on a panorama image gallery. As displaying all panoramas of the route would result in too much displayed pictures only the panoramas of decision-points are provided. Another hypothesis is that users can memorize some of the upcoming turns and therefore will get faster to the end of the path. To simplify orientation within passed and future locations users can tag locations in the preview list as "already seen".

Additionally to these features a re-localize functionality is provided. The main thought is that if users feel lost or insecure about their current position there has to be the possibility to display the imminent way instruction. Whenever they want to re-localize themselves users lift the phone as if they would like to take a photo. The motivation is that the fact that a vision-based system has to "see" its actual environment in order to give feedback about the user's location is comprehensible and thus easy to memorize. As highlighted above users only see panoramas of decision-points in the manual mode. But a re-localization request renders the panoramas and the corresponding way instructions at users' locations, even if they are not at a decision-point location.

A further concept developed during this thesis considers the manual mode, too. After an initial study the question arose where users would find getting way instructions important. To track the subjective preferences the manual mode is condensed. Henceforth, it only offers the possibility to request instruction presentations of the actual user position. The user has no knowledge about the route and is motivated to request updated instructions only if necessary. The second version of the manual mode aims at gaining insights about locations being important to a majority of users. The findings may then be used to improve future path definitions according to users' preferences.

Chapter 4.

Implementation

This chapter lists in detail the changes made to the application developed by Soulard [47]. During the working time of this thesis three different instruction modes, a map preview module and some miscellaneous extensions were created. At first the starting point of this thesis is explained. Afterwards the instruction modes, namely the fully automatic mode, the decision-point mode and the manual mode are outlined. Then follows a description of the map preview module. The chapter ends with a section listing extensions the modules for server-client communication, path editing or creating and logging.

4.1. Starting Point

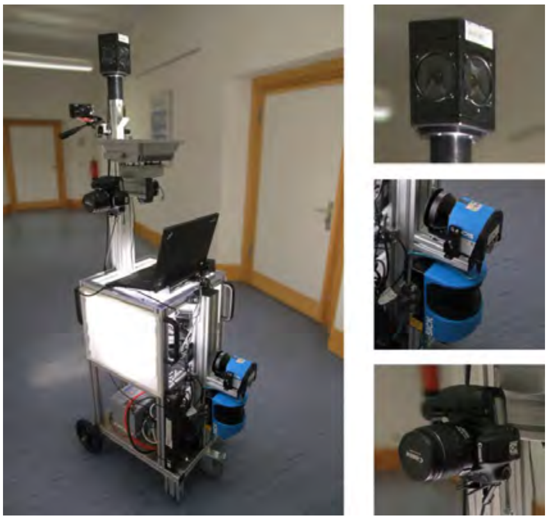
In this section a short summary of the base application is given. Only the parts relevant to this thesis are highlighted. For a description in full detail of all features see Soulard's master's thesis [47]. As the system depends on a client-server architecture [45] first the client components will be described and subsequently the server features and the client-server communication will be laid out. Note that this is a summary of already existing software which serves as a starting point for this thesis.

The main component of the *HallwayView* application is a view layer rendering 360° panorama scenes. Figure 4.1a shows the camera setting of the capturing trolley. The application renders five horizontal panorama images spanning a whole circle around the user's point of view into an Android SurfaceView[1]. The images are mapped as textures onto a sphere surrounding the camera conveying a three-dimensional impression.

Figure 4.1b shows the panorama view of the client application. The main element is the red arrow displaying the walking direction. Moreover there is information about the route and the walking speed. In the upper left corner a button for connection handling is provided and the button in

[1] `http://developer.android.com/reference/android/view/SurfaceView.html`
(last accessed: 03/29/2013)

the upper right corner switches between the panorama and an AR view. Users may rotate the panorama view around their current point of view by swiping horizontally over the screen.



(a) The custom-built trolley recoding panorama images based on a Ladybug3[2] setting. Source: [48].

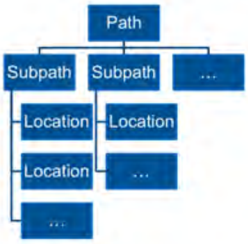
(b) Panorama view of the HallwayView application.

Figure 4.1.: Impressions on the application.

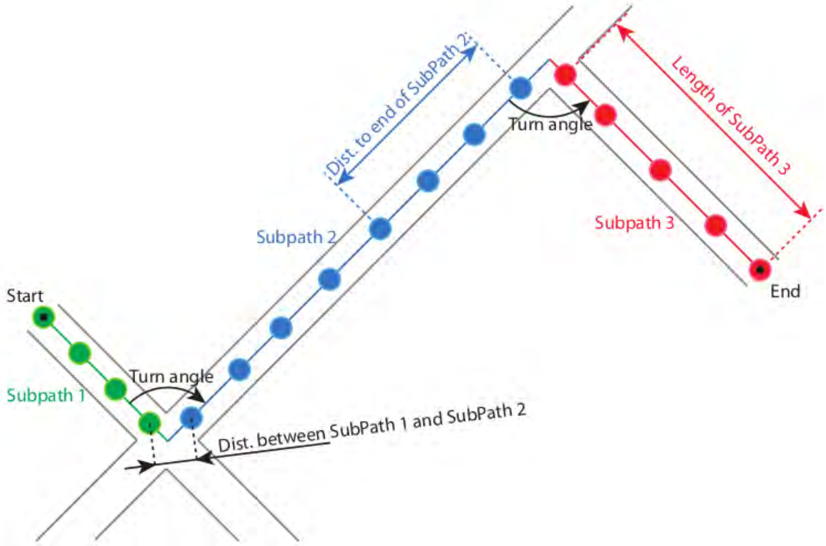
All information about the route (except the panorama images) is stored and managed on a server application running on another mobile phone. The experimenter carrying this second phone sends instruction updates of the user's current location to the *HallwayView* application. There the updated route information is rendered. The server application also controls the logging functionality and some other features of the client.

The route information is held in a hierarchical path structure which consists of a root node called *Path*, child nodes called *SubPath* and leaves called *Locations* (Figure 4.2a). A *SubPath* is a collection of *Locations* between two turns. Each *SubPath* has an orientation and an angle. The orientation gives the overall direction of the *SubPath* while the angle denotes the difference in degrees between the *SubPath* itself and the following one. As most buildings are build rather orthogonal *SubPath*s are mostly straight and angles are either 0°, 90° or -90°. At the end of a *SubPath* with angle 0° the user would be informed to go ahead, an angle of 90° would result in the instruction to turn right and a left turn is given by an angle of -90° (Figure 4.2b). Each *Location* knows its coordinates and thus can calculate the Euclidean distance to any other *Location*. Accordingly the length for each *SubPath* may be determined and from each *Location* the distance to the beginning and the end of the *SubPath* is known. This is especially useful if one wants to know the distance to the next turn location. Paths creation and editing is treated in Section 4.4.2.

[2] http://www.ptgrey.com/products/ladybug3/ladybug3_360_video_camera.asp (last accessed: 03/29/2013)



(a) Hierarchical path structure.



(b) *SubPath* to *SubPath* relation.

Figure 4.2.: Path and path elements organization. Source: [47].

4.2. Instruction Modes

The goal of this thesis is to evaluate instruction presentations for indoor navigation in order to improve the route directions and optimize the user interaction. To test different conditions several application modes are implemented into the application introduced in Section 4.1. In both evaluation studies users evaluate three different instruction modes: a *fully automatic*, a *decision-point* and a *manual* condition. Thus, this thesis' work includes the restructuring of the existing application so that information relevant to a distinct instruction mode can easily be hidden or displayed. A start screen offers the possibility to choose between the modes Demo, A, B and C to all users (Figure 4.3). After clicking on one of the alternatives the appropriate condition is initialized automatically. It is necessary to mention that the Demo mode resembles the fully automatic mode on a very short path. It is especially designed for introducing users to the functionalities of the system.

4.2.1. Fully Automatic Mode

The fully automatic mode displays route instructions at every location of the path. After the experimenter, who definitely knows where the user is at the moment, selects a new location at

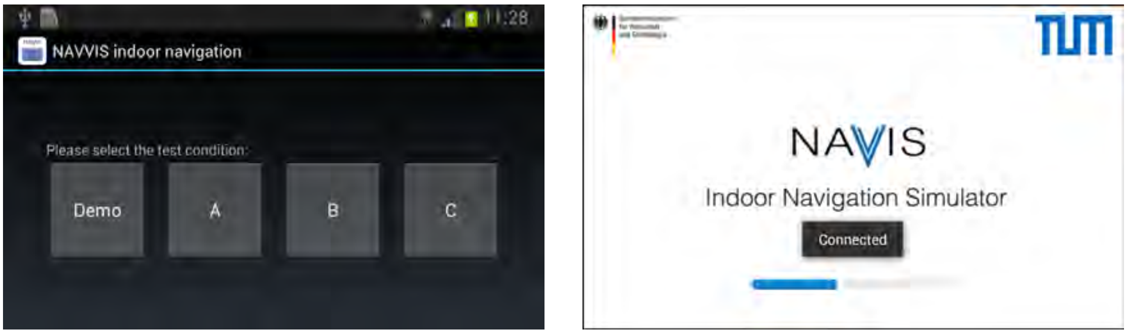


Figure 4.3.: The initial start and loading screens.

the server side, the client panorama scene and the included instruction presentations are updated (Figure 4.4). This includes re-rendering of the image set belonging to the received location, displaying the direction arrow according to the updated way instruction and replacing icons and markers in order to keep the interface elements consistent. As its name says this mode guides the user fully automatically to the goal. However, users have the possibility to rotate the virtual viewport by swiping horizontally over the screen. Hence, users are able to look around in the virtual scene. Note that this mode was not developed during this thesis. It is adopted as it stands from Soulard's thesis [47] with minor but necessary changes.

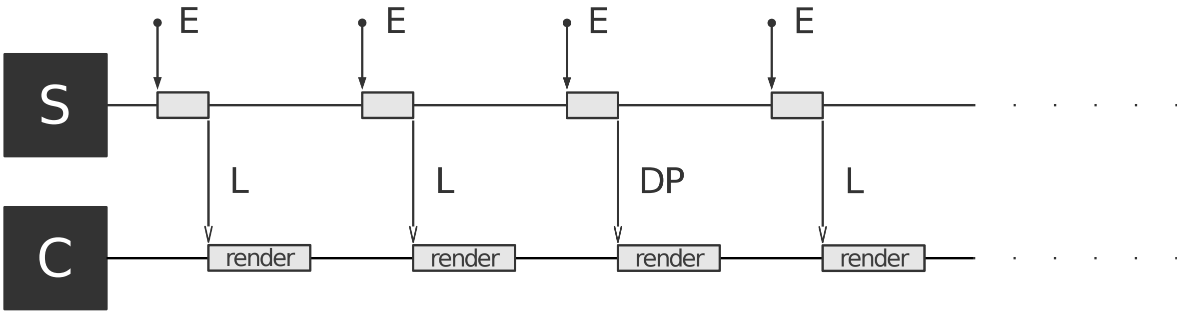


Figure 4.4.: Sequence diagram for the server-client interaction in the fully automatic mode. The experimenter (E) selects a location (L,DP) and sends it to the client. There decision-points (DP) and normal locations (L) and instructions are rendered.

4.2.2. Decision-Point Mode

The concepts of this thesis make a major change to Soulard's work [47] necessary. While the fully automatic mode displays all locations of a path, the decision-point mode displays new instructions only at turns or forks. In the scope of this thesis the logic of the fully automatic mode is adopted and then enhanced by the features of the decision-point concept. This means that the instruction presentations, e.g. the direction arrow, stay the same but the frequency of instruction updates is changed. The decision-point mode takes care to display new panoramas and instructions only after passing the previous decision-point. The experimenter's behavior does not change. Location information is continuously sent to the client application but only decision-point information is rendered (Figure 4.5). While users walk along a hallway they constantly see the panorama and turn instruction of the location at the end of the hallway. As soon as they have passed this point the panorama and the instruction presentations are updated and indicate the maneuver at the next decision-point.

The application contained little bugs in the version for the first study. The distance information to the next decision-point always was "0m". This was because the distance between the displayed panorama and the next decision-point, which are identical in this case, was calculated. In the version for the second study this bug is removed and the distance between the continuously received, but *not* rendered, locations is determined. Thus, while walking down a hallway the distance information showed continuously decremented measures. Moreover, the finish flag rendered as soon as the user reaches the destination was shown too early in the first version. It was displayed when users passed the last decision-point and entered the final hallway. Hence, they saw the finish flag before having reached the goal. Also, this is improved in the second version of the decision-point mode and participants of the second study can not observe the black-white checkered flag until the last location of a path is reached.

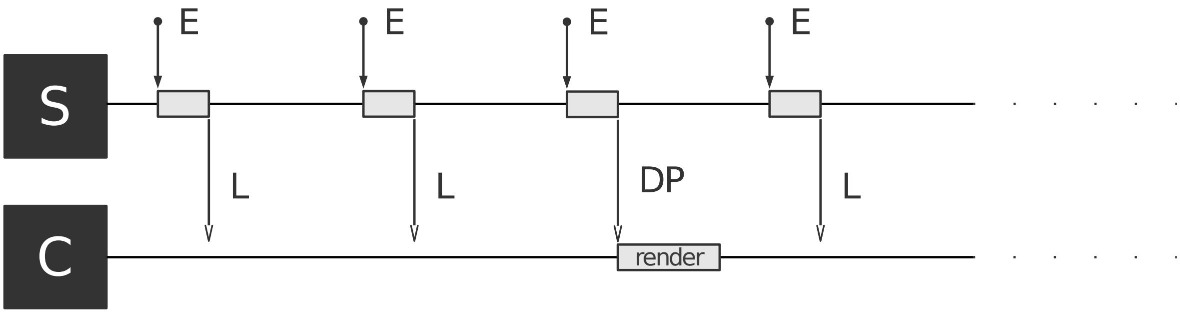


Figure 4.5.: In the decision-point mode the sequence of server-client interactions is as follows: The experimenter (E) selects locations (L,DP) and sends them to the client. There only the decision-point instructions and panoramas are rendered.

4.2.3. Manual Mode

The manual mode conceptually allows users to select and update instruction presentations manually. It is possible to select and display locations from a panorama gallery or to sift step-by-step from one instruction to the other. If users get lost in the diversity of panoramas and instructions they have the possibility to request an automatic re-localization. This feature, thought to be a last resort, then displays the panorama and the corresponding instruction of the user's location. Locations displayed are in general decision-points only. But the re-localization information may also be an arbitrary location of the path. The experimenter acts equally to the two above highlighted modes and provides the manual mode with continuously update information about the user's location. All manual requests from the client are answered automatically by the server without interaction by the experimenter (Section 4.4.1). Rendering of panoramas and instructions is restricted to update events resulting from manual request. Thus, the continuously sent location informations are not noticed by users (Figure 4.6).

At first this section highlights the panorama gallery, called *PathPreview*, with its interface elements and interaction possibilities. Afterwards, the implementation of the general interaction concepts are outlined.

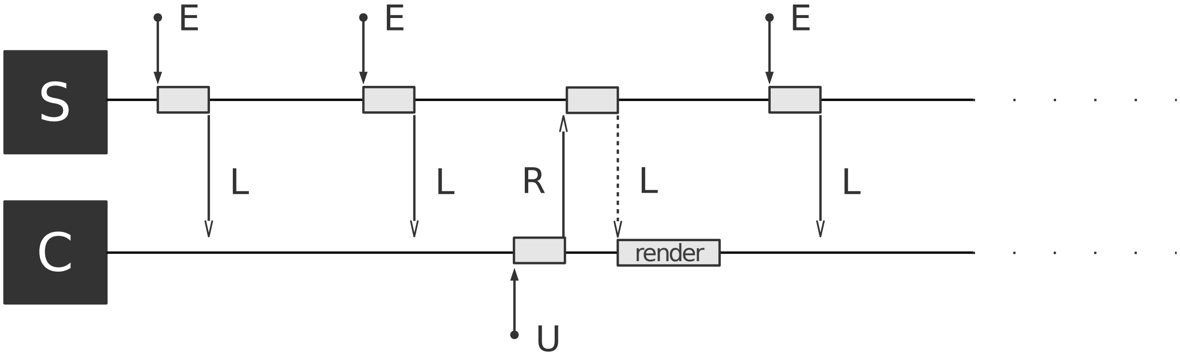


Figure 4.6.: Sequence diagram illustrating the interactions between server and client in the manual mode. Locations (L) sent by the experimenter (E) are not rendered on client side. Only locations sent in response to requests (R) by the user (U) update the panorama and instruction presentation.

Path Preview

Interface The path preview is designed as a module put into an extra class. It offers an interface that can be hidden at the right edge of the screen. The view is implemented as a SlidingDrawer[3] a view group included in Android's API. This kind of view hides all contained content but provides a handle to pull the content back to the screen. Figure 4.7 illustrates this behavior with screenshots of the application. After the slider is fully expanded an additional view displays a preview of the selected panorama in the list at the right. The standard SlidingDrawer cannot be configured to wrap its contents. After pulling the slider onto the screen it maximizes automatically and occupies the whole screen. A solution for a *WrappingSlidingDrawer* may be found on the platform Stackoverflow[4].

[3] `http://developer.android.com/reference/android/widget/SlidingDrawer.html` (last accessed: 03/29/2013)

[4] `http://stackoverflow.com/questions/3654492/android-can-height-of-slidingdrawer-be-set-with-wrap-content#4265553` (last accessed: 03/29/2013)

Figure 4.7.: Path preview gallery slides into view.

Data management The data of the list view at the right is managed by a class called *GalleryImageAdapter*. This is an extension of the API class ArrayAdapter[5]. The adapter is attached to a ListView[6] and each time a list is loaded it calls the *getView(. . .)* method on the adapter. The *GalleryImageAdapter* works on *ImageDetail* items. The *ImageDetail* class takes care for the data stored with an image. Beside others the image identifier, the image file path and the image bitmap, it stores if the image is currently selected or if it has been tagged as "already seen". Before returning a view in the *getView()* method the *GalleryImageAdapter* checks the requested item for this last two properties and returns selected items with a gray border and tagged items marked with a green dot (see Figure 4.9, right image). The images from the Ladybug-dataset used in the application have a resolution of 512x512 pixels and are rotated by 90° thus they have to be scaled and re-rotated before returning them to the ListView. Once requested, item data is cached for sake of performance. Due to this modularized structure the code stays clear and maintainable.

Preview Interaction There are three interaction possibilities when the path preview is opened. By tapping on the big panorama thumbnail in the middle of the screen users select the displayed location which is then rendered as a 360° panorama including all route instructions. Therefore the *HallwayView* module implements an interface provided by the *PathPreview* module called *PathPreviewListener*. Each time a preview is selected the listener is notified via the *onPreviewSelected(int id)* method and is then able to do further work with the given image identifier. To select another preview users tap on one of the little thumbnails at the right. The list then scrolls automatically to center the selected thumbnail horizontally. Figure 4.8 shows that the transition between the old and the newly selected preview images is done by a cross-fade animation. Of course users may also scroll the images at the right by swiping up or down on the list view. The end of the scrolling animation was enhanced by a center-horizontally effect, too. After the scrolling is completely done the item in the middle is selected. The list is not centered if the first or the last item is selected or if it is scrolled to the beginning or end. In one of this cases only the gray border around the items indicate the currently selected preview. A further visual gimmick would be to move the triangular arrow of the preview box up or down to match the selected item in cases when the list is not centered.

[5] `http://developer.android.com/reference/android/widget/ArrayAdapter.html` (last accessed: 03/31/2013)

[6] `http://developer.android.com/reference/android/widget/ListView.html` (last accessed: 03/31/2013)

Figure 4.8.: Selected thumbnail is centered horizontally.

General Interaction Features

While the path preview is closed the manual mode provides two additional interaction possibilities: swipe and confirm. In order to be able to react on user gestures the *PathPreview* module extends the Android API class SimpleOnGestureListener[7]. Besides others it offers the methods *onFling(. . .)* and *onDoubleTap(. . .)*.

Swipe The manual change of panoramas is triggered by a swipe or fling gesture. As soon as a fling gesture is detected it is tested if it was long enough, i.e. the distance (in pixels) between the start and the end of the gesture exceeds a certain threshold, and if it was fast enough, i.e. if the former mentioned distance was passed within a given time threshold. These threshold tests avoid unintended gestures when the user accidentally touches the screen. If the system detects a gesture it selects the previous or next (depends on the gesture direction) panorama automatically. The rendering of the selected panorama is triggered via the above highlighted listener pattern. If there is no previous or next image the user is notified by a short Toast[8] message. The manual changes update the preview list even if it is not visible so that if users expand the preview again the currently selected item is focused in the list view.

Confirm An alternative to swipe to the next location is to double tap on the displayed panorama. This automatically selects the imminent location and confirms that the double-tapped location was "already seen". Figure 4.9 illustrates from left to right how the double-tapped panorama is switched to the next panorama on the path and how the checked items are rendered in the preview list when the *PathPreview* is expanded.

Re-localize In order to trigger a re-localization users lift the phone as if they would like to take a photo. This metaphor (see Appendix A) is based on the fact that a vision-based system has to "see" the user's actual environment before being able to determine the user's location. In fact the functionality is realized by listening to the inbuilt gravity sensor of the mobile device. Whenever the inclination of the phone exceeds a certain threshold the client automatically sends a

[7] `http://developer.android.com/reference/android/view/GestureDetector.SimpleOnGestureListener.html` (last accessed: 03/30/2013)
[8] `http://developer.android.com/reference/android/widget/Toast.html` (last accessed: 03/30/2013)

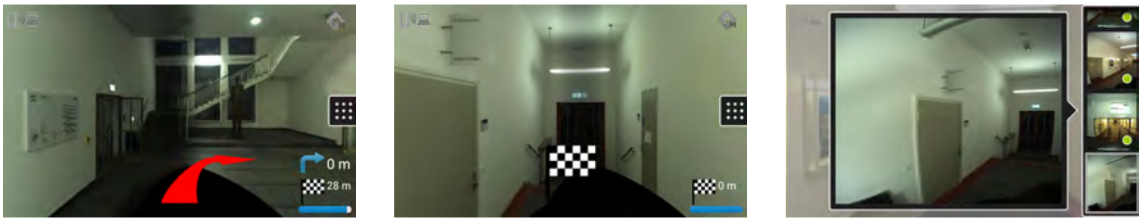


Figure 4.9.: Double-tapping automatically renders the imminent panorama and marks items as "already seen".

message, requesting the current location information, to the server. There the corresponding route instruction for the requested location is extracted from the path definition and automatically sent back to the client where the panorama and the instructions are updated accordingly (Figure 4.10).

Figure 4.10.: The re-localization process: The user sees the last decision-point panorama, goes on and enters the building, there the future way is unclear so the user lifts the phone to trigger the re-localization and receives the location information of his or her current position.

4.2.4. Instruction Presentation in the Follow-Up Study

The instruction presentation in the follow-up study differed significantly from the initial version of the presentation modes. The fully automatic mode and the decision-point mode are extended by the *MapPreview* (see Section 4.3) and minor bugs as highlighted in Section 4.2.2 are fixed. The manual mode is heavily restructured. The *PathPreview* is disabled and the interaction possibilities for manually changing the panorama are removed. The interface is condensed to the VR scene with a single button. Touching this button in fact triggered a re-localization and resulted in a updated panorama with corresponding instructions of the actual position of the user.

The follow-up study did not aim at comparing the manual mode to the other two modes. The manual mode is simply used to track were users find getting new way instructions important. The interface of the manual mode shows a counter displaying the number of requests. To keep the requests for updated panoramas and instructions at a necessary minimum and to motivate users not to trigger unnecessary requests the counter changes its color from green over orange to red.

This is a subtle way to tell users if their request count is still reasonable or if they are heading towards "too much" request.

4.3. Map Presentation

The findings from the first study (see Section 5.4) approve the fact that users need additional help when not guided by a fully automatic system. Here we refer to the problems users had with orientating in the zigzag of path B, to difficulties with already seen and future panoramas in the manual mode or the fact that users did not seem to know where they are after having reached the goal. Related work [45] found that a map view is not necessarily needed, however, the highlighted problems are reason enough to integrate a map concept as it is explained in Section 3.2. In what follows the implementation of this concept is outlined.

The map extension is based on the Google Maps API for Android[9]. The extensive programming interface offers enough possibilities to create custom map presentations. With version 2 the API was further improved for mobile use (using vector tiles and reducing bandwidth by improved caching) but getting started with the implementation was made more complex as the process of obtaining a developer key and integrating it into the Android project seems unnecessarily complicated. This section is about the elements displayed on the map, the user interaction and some special problems like making map tiles available offline.

4.3.1. Map Elements

The elements displayed on the customized map are the user's location in the building, the passed and future path along the hallways and the next decision-point on the way from the user's actual position to the end of the path. The map elements are not managed in the class *MapPreview* directly, all data is held by and all functions are delegated to the class *LocationOverlay*.

User Position The user position is marked with an arrow icon. Additionally to the location the arrow illustrates the user's viewing direction (see Figure 4.11d). The current location of the user is derived from the location updates sent by the experimenter application. In Section 4.1 the NAVVIS project *TUMIndoor* is mentioned. The image dataset[10] resulting from this project assigns geographic coordinates to all recorded images. Section 4.3.3 describes how the different coordinate systems, namely the WGS84 system for Google Maps and the UTM system for the NAVVIS reference data, are matched. As new location updates are received by the client, the included location information is extracted and passed on to the *MapPreview*. There the coordinates

[9] `https://developers.google.com/maps/documentation/android/` (last accessed: 03/30/2013)
[10] `http://navvis.de/dataset/` (last accessed: 03/31/2013)

are transformed and the user marker update is delegated to the *LocationOverlay*. This class makes use of the Google Maps API to remove the old marker and to set a new marker to the new position. In the meantime the viewport of the map is updated. A translate animation to the new location is performed so that users see the map "sliding" to the current position. These constant viewport updates assure that the user marker is always centered and visible.

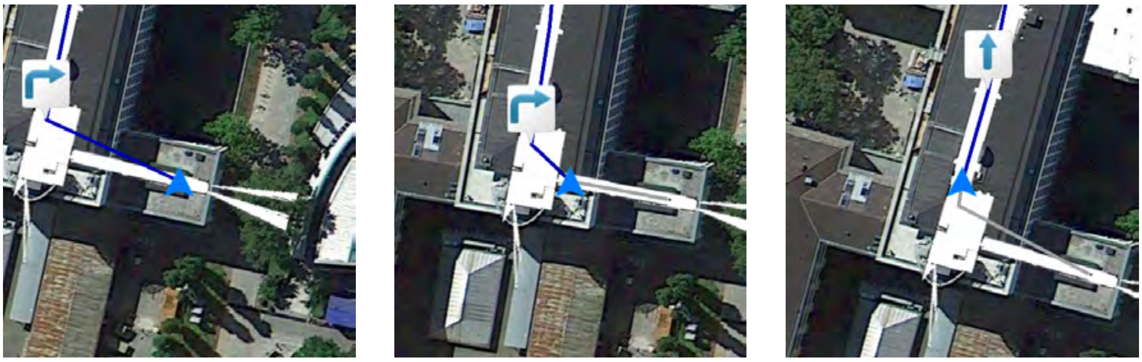
(a) straight ahead (b) turn left (c) turn right (d) user (e) finish

Figure 4.11.: Markers used by the *LocationOverlay*.

Path Polygon The selected route is displayed as a polygon line. The vertices of this line are the first locations of each path segment (*SubPath*) and the last location of the path. The information about the exact positions of these vertices is included in the initial message received from the server. As this message arrives, the client passes the identifiers and the coordinate information of the vertex locations to the *MapPreview*. Again the coordinates have to be mapped and afterwards the path is rendered on the *LocationOverlay*. The rendering process includes adding the user marker to the initial position and a black and white checkered race finish flag, representing the end of the path (see Figure 4.11e).

The passed and future path are colored differently. The portion of the route already passed by the user is drawn gray while the path to come is colored blue. The *LocationOverlay* manages passed and future vertices and in fact draws two path polygons: a gray and a blue one. As the panorama locations are not always on a straight line between two path vertices the connection between the continuously updated user marker and the last vertex (decision-point) would sometimes span across walls and no-hallway areas. To overcome this and to give a better impression of the already walked path all locations between the last decision-point and the current user position are stored and included in the route polygon. As this would imply storing many locations as the users passes on only the locations between the very last decision-point and the user position are stored. As soon as the users passes a decision-point the intermediate locations between the last but one decision-point and the just passed decision-point are discarded. The difference between storing intermediate locations or simply connecting vertices with a straight line is illustrated in figure 4.12.

Next Turn In order to give users the possibility to relate the displayed panorama of the next decision-point with a real location in the building a marker at the next (seen from the user's position) vertex is displayed. This was thought especially useful in the *decision-point* mode. The



(a) Polygon spans some walls

(b) Passed path is colored gray and intermediate locations are used

(c) After passing a decision-point the intermediate locations are discarded

Figure 4.12.: Path polygon updates.

marker icon varies according to the direction of the next turn. Figure 4.11 shows possible turn icons and figure 4.12 gives an impression how the turn marker is embedded in the map view. Again updates to the markers are received by the *MapPreview* but delegated to the *LocationOverlay* which manages all elements displayed on the map.

4.3.2. Map Interaction

The following section lists the different possibilities to interact with the map view. It is described in detail how the map may be rotated, how it can be maximized and minimized and how the viewport may be panned and zoomed.

Rotation Section 3.2 explains the advantages of self-rotating maps. This paragraph highlights how this concept of map rotation is implemented. The information about the orientation of the phone is received from a gravity sensor which is nowadays built into the majority of mobile phones. The Android API provides a SensorManager[11] which takes care for the different inbuilt sensors. An application interested in sensor updates simply implements the SensorEventListener[12] which is then notified on sensor and accuracy changes. Additionally the necessary calculations to map the phone's to the world's coordinate system are done by the SensorManager, too. Thus the rotation, inclination and orientation of the phone may be received easily. In his master's thesis Soulard describes in detail the use of the SensorManager in the context of the *HallwayView* application [47]. To match the calculated orientation angle α to the orientation of the map

[11] `http://developer.android.com/reference/android/hardware/SensorManager.html` (last accessed: 03/31/2013)

[12] `http://developer.android.com/reference/android/hardware/SensorEventListener.html` (last accessed: 03/31/2013)

respectively the TUM building it has to be rotated clockwise by 90°. As in this case α is in the range $]-360;0]$ the transformation is the following:

$$\alpha_{TU} = (270 - \alpha)\ \%\ 360$$

Sensor updates are received about every 10 milliseconds. As the existing logic of *HallwayView* should not be altered, which would mean to change the update delay of the SensorManager, another means to prevent map orientation jitter is realized. Instead of updating the map's orientation with every sensor update the angle of the last map update is stored and further sensor updates are ignored until they differ by 10° to the stored value.

Maximize & Minimize The second interaction concept concerning the map view is the possibility to enlarge the map area which is by default displayed in a 300x300px container in the upper right corner of the screen. A maximized map shows more route details and provides more reference-points in the environment. To provide an overall nice look and feel in the application the resizing of the map is animated.

The general gesture to maximize and then minimized the map again is to double tap the map area. As double-tapping the map is an event reserved by the Google API to zoom the map an invisible overlay consuming the double-tap event is placed above the map area. This area simply listens on touch events inside its borders. A realization like in section 4.2.3 where double-tap events are used is not possible as the Google Map module completely consumes these events. Thus double-tap events are simulated by two single-taps. If two consecutive taps follow each other within a certain time and are close to each other a double-tap is assumed and the map size is toggled. Let e_1 and e_2 be the two single-tap events and t_t, t_x, t_y are the thresholds for $time$, x and y deviation then a double-tap action is triggered if the following conditions are met:

$$e_2.time - e_1.time < t_t$$
$$|e_1.x - e_2.x| < t_x$$
$$|e_1.y - e_2.y| < t_y$$

It is highlighted above that the resizing of the map should be animated. Hereby the problem is the animating the size of the Google Map container seems not possible. Using the Android animation package[13] to scale the map view results indeed in an animated scale of the view container while the map content is not animated. Hence another workaround is needed. In advance to the maximize animation the map is resized to match the screen size but the left and lower border of the map container stay the same. So the additional content is clipped by the screen borders and the user

[13] `http://developer.android.com/reference/android/view/animation/Animation.html` (last accessed: 03/31/2013)

does not notice the resizing (check Figure 4.13 for clarification). In a second step the position of the map view is animated which seems not to evolve any problems. The view is translated down and left until it fills the whole screen. When minimizing the map again this process is reversed. First the maximized map view is moved to the upper right corner until a 300x300 pixel area remains on the screen. Then the map is resized and minified so that upcoming position updates do not require additional calculations. This would be the case if the map area stays maximized with a part off-screen but all location changes and camera updates would still be relative to the 300x300 pixel area.

Figure 4.13.: Illustration of the maximize animation.

Pan & (Zoom) The default pan and zoom interactions are possible with the *MapPreview*, too. Both are delegated to and managed by the Google Maps API. There are two unsolved problems. The pan interaction triggers a double-tap if users swipe too fast in the map area. In this case the workaround for double-taps mentioned above is not working as expected. Two consecutive swipe gestures intended to pan the map trigger two touch events which fulfill the mentioned conditions. Until now there is no solution for this because no “finger-up” event is triggered, only the “finger-down” events are reported. This makes it impossible to distinguish between pan and double-tap gestures. The second problem concerns the zoom interaction. At the moment it is disabled because there is no consistent Internet connection in the hallways of the TUM building. Thus the map contents for one specific zoom level and a limited area have been downloaded. As zooming would require other zoom levels it is not available at the moment. See the next section for a more detailed description of the solution for this problem.

A comprehensive WLAN coverage can not be taken for granted in all application areas of an indoor navigation system. The presented solution however is applicable to any location covered by the Google satellite images. Additionally integrating open-source projects like OpenStreetMap[14] is possible. Even more sophisticated ideas like providing customized map data, e.g. heatmaps of crowded corridors, are conceivable.

4.3.3. Map Challenges and Solutions

In the former paragraph it was already told that it was necessary to make the map contents, i.e. the map tiles, available offline for our survey prototype. This section first describes how this problem was solved, then explains how the different coordinate systems included in the *MapPreview* module were mapped and third it tells about how problematic it can be to stack two SurfaceViews[15].

Offline Availability Normally the Google Maps module receives all necessary data like satellite or schematic map images via an Internet connection from the Google servers. Unfortunately not all hallways in which we conducted our study are covered by a WiFi network. Thus the client application has only limited access to the Internet for updating the map view. This led to the decision to cache the map contents on the SD card of the device and later feed the map with data from a custom TileProvider[16]. Providing tiles of a map is a common concept to save bandwidth in map applications. Instead of transmitting one large map image the map is divided into several rectangles (or tiles) and only the images of the currently visible tiles are transmitted. Each zoom level of a map then consists of a different number of tiles. With Google Maps the tiles have a size of 256x256 pixels. At zoom level 0 there is exactly one tile showing the whole earth. The next zoom level divides the first image into 4 equally sized parts. Zoom level 2 would then consist of 16 tiles. The nth zoom level then has 2^{2^n} tiles (current maximum with Google is $n = 19$). Figure 4.14 illustrates the correlation between zoom level and number of tiles.

The download of the relevant tiles for the TUM building and its environment was automated by a small shell script. The programm cURL[17] was used to fetch the map tiles from the Google servers. The parameters for the x and y coordinate of the URL

[14] `http://www.openstreetmap.org/`, last accessed: 04/14/2013
[15] `http://developer.android.com/reference/android/view/SurfaceView.html` (last accessed: 03/29/2013)
[16] `http://developer.android.com/reference/com/google/android/gms/maps/model/TileProvider.html` (last accessed: 04/01/2013)
[17] `http://curl.haxx.se` (last accessed: 04/01/2013)

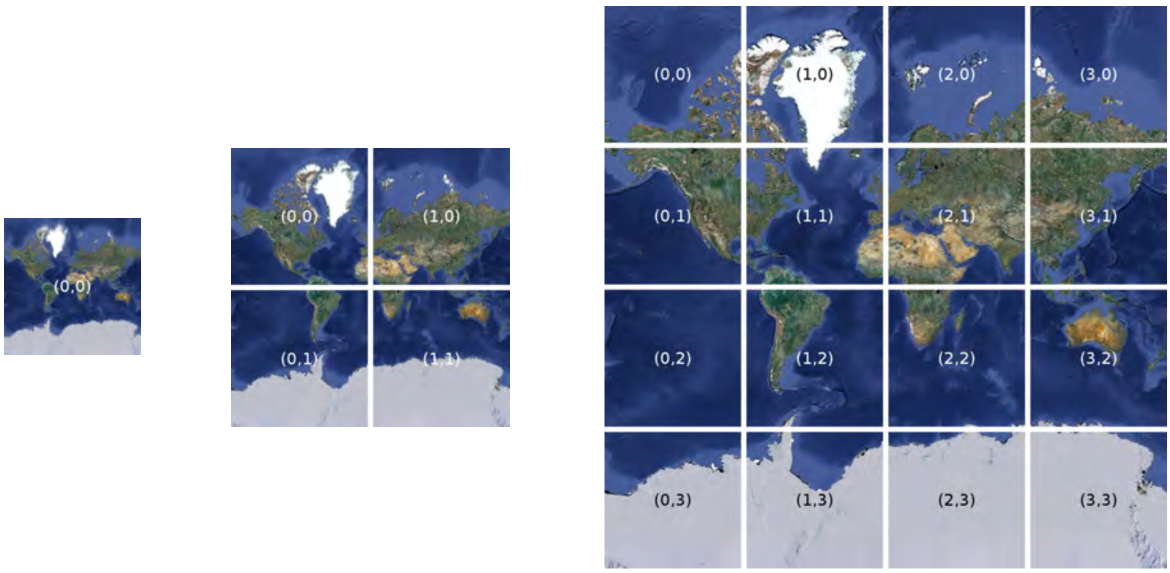
https://khms1.google.com/kh/v=126&src=app&**x=0**&**y=1**&z=1 are iterated in two FOR-loops. Fortunately cURL offers the possibility to provide a proxy address as Google blocks automated request after a while. The download script is not yet capable of switching the proxy option on its own as soon as the requests are blocked. Hence proxy servers have to be searched and the option needs to be updated manually. This is the reason why only zoom level 19 is provided in our implementation. Of course a more mature version would include offline tiles for different zoom levels. For sake of usability the downloaded tile images are saved in a $zoomlevel \rightarrow x \rightarrow y.jpg$ folder hierarchy.

After downloading the map tiles the floorplan images that were formerly downloaded from NAVVIS servers by the applications had to be cached for offline use as well. Some of the floorplan images have a file size greater than 10 megabytes (which have to be held in memory), so it was decided to create tiles for them, too. Starting with the high resolution floorplan image the creation process is organized in three main steps:

1) Image Anchor Point Matching In order to correctly locate the image one has to know the geographic coordinates of one point in the image

2) Image Scale and Rotation After the image is correctly located it has to be scaled according to the zoom level and in case it is not matching the building orientation it has to be rotated

3) Tile Export According to the Google coordinates for tiles the correct portions of the high resolution image have to be exported



(a) zoom 0 (b) zoom 1 (c) zoom 2

Figure 4.14.: Tiles for different zoom levels.

Image Anchor Point Matching

The NAVVIS TUM Indoor Viewer[18] provides for each floorplan image the information[19] necessary for these operations. For each image the width w and height h in pixels are known. Each image has a certain resolution r defining how many meters correspond to one pixel. Furthermore the spherical mercator coordinates of the image's origin o_x, o_y are known. The anchor point itself is defined relative to this origin. First the independent anchor point coordinates a_u, a_v in the range $[0; 1]$ are determined before the anchor point in pixels a_x, a_y can be calculated.

$$a_u = \frac{-o_x}{r * (w-1)}, a_x = w * a_u$$
$$a_v = \frac{-o_y}{r * (h-1)}, a_y = h * a_v$$

Image Scale and Rotation

After the image is positioned so that anchor point and the image origin match it has to be scaled and rotated. The new scale depends on the zoom level we are creating tiles for. This means we have to know how many meters per pixel m_{px_l} exist at latitude l of the building. Then we can derive the new width in pixels w_{px} for a certain zoom level z from the original width in meters w_m. In the following equations px_e defines the pixels for all tiles juxtaposed at the equator at zoom z and px_l stands for pixels at latitude l.

$$px_e = 256 * 2^z, px_l = p_e * \cos(l)$$
$$m_{px_l} = \frac{40075017^{20}}{px_l}$$
$$w_m = w_{px} * r$$
$$w_{px} = \frac{w_m}{w_{px_l}}$$

The initial high resolution image is then scaled to the width w_{px}. Additionally it is rotated by angle α given in the NAVVIS data[19] to match the building orientation. It is important to note that the scale as well as the rotation transformations are made relatively to the above calculated anchor point.

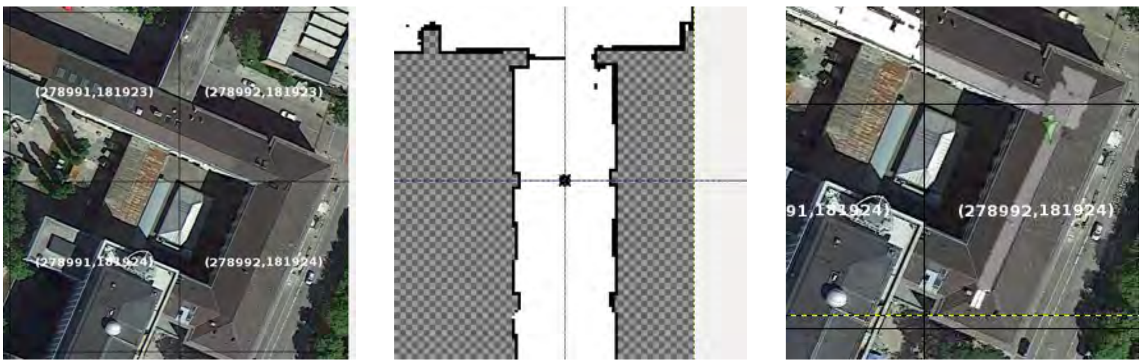
[18] http://navvis.de/view/ (last accessed: 04/01/2013)
[19] http://www.navvis.de/view/data/datasets-dev.js (last accessed: 04/01/2013)
[20] WGS defines equatorial circumference as 40075017 meters

Tile Export

Simply exporting the scaled and rotated image as 256x256 pixel tiles is not enough. One has to make sure that each exported portion matches the Google coordinate system. That is why we followed a rather visual creation process including the above mentioned transformations. To do this GIMP was used. First a grid of map tiles and tile labels (naming the tiles with their x/y coordinates) was created (see Figure 4.15a). Then the anchor point in the floorplan image was marked with a small dot (Figure 4.15b). Afterwards the scale and rotation transformations are applied. The manipulated floorplan image has then to be matched to the underlying tiles. As the image origin o is given in spherical mercator coordinates it has to be transformed to lat, lon before further use with Google is possible.

$$
\begin{aligned}
lon &= \frac{o_x * 180}{20037508.5\text{[21]}} \\
lat' &= \frac{o_y * 180}{20037508.5} \\
lat &= \frac{180}{\pi} * (2 * atan(e^{\frac{lat' * \pi}{180}}) - \frac{\pi}{2})
\end{aligned}
$$

These coordinates are then provided to the Google Maps search. A screenshot from the resulting screen at zoom level z is taken and put into the GIMP project. After translating the screenshot until it matches the included tile images the green marker denotes the position of the floorplan image anchor point. When the floorplan image is moved so that the formerly marked anchor point matches the green marker the floorplan tiles are ready to export (see Figure 4.15c). We only have to extract portions of the floorplan image fitting the Google tiles marked by the black lines (Figure 4.16).

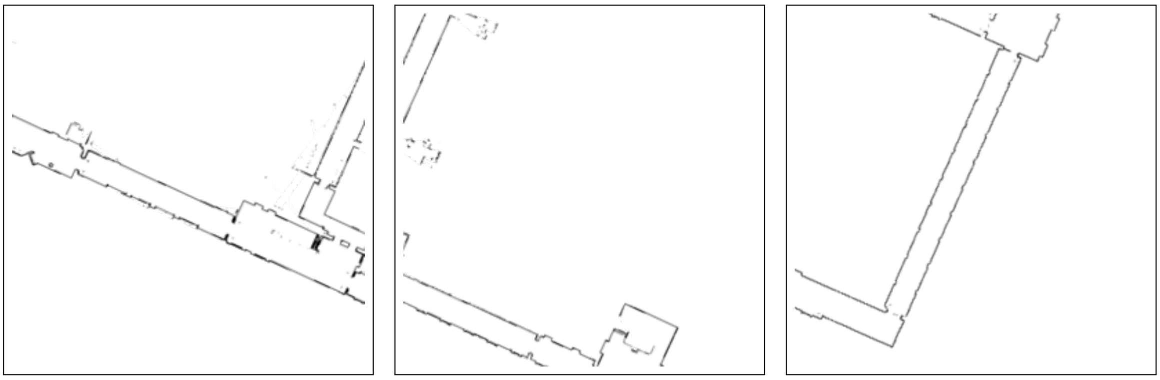


(a) Tile grid with coordinate labels. (b) Anchor point marked in floorplan image. (c) Origin mapped to anchor point.

Figure 4.15.: Tile creation steps.

[21] half of the WGS equatorial circumference

The map tiles are provided to the Google Map module in our *MapPreview* by a custom TileProvider called *CustomTileProvider*. This abstract class requests its sub-classes to implement the method *getTileFilename(x, y, zoom)*. This offers the possibility of re-using code while different tile images, e.g. the offline map and floorplan tiles which are stored in different folders, may be provided. The *CustomTileProvider*s are attached to the map module as soon as it is initiated by the *MapPreview*.



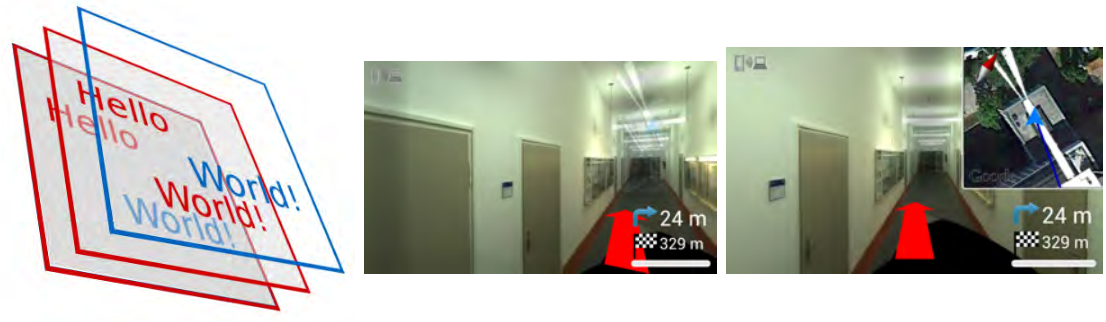
(a) (278991,181923) (b) (278992,181923) (c) (278992,181924)

Figure 4.16.: Resulting floorplan tiles annotated with their tile coordinates after the export step.

Coordinate System Mapping We saw in the former paragraph about the offline availability of the map content that Google and the NAVVIS project use different coordinates systems. Whereas Google's map presentation is based on the WGS84 system, NAVVIS uses spherical mercator coordinates. The NAVVIS image database is further divided into building floors and sections and each of those sub-datasets has its own orientation and point scale. The point scale relates metric units to units in the spherical mercator projection according to the latitude. Each sub-dataset provides a method taking into account these parameters to calculate the WGS84 coordinates for its items. Hence for each vertex or user position update the dataset containing the updated location has to be identified in order to get the "correct" coordinates. At the moment there are only five sub-datasets so this is not a performance killer but in a bigger database it would be advantageous to put all sub-dataset to a common denominator.

SurfaceView z-Order This paragraph describes the challenge of stacking two SurfaceView layers above each other. Normally the behavior of stacked layers is that the last added layer is rendered above all formerly added layers. One will observe in Figure 4.17a that the rendered text (displayed on the lowest layer) is "Hello World!" with a red "Hello" and a blue "World!" while the phrase is completely red in the middle layer. That is because the blue contents in the top-most layer cover the contents of the lower layers.

With stacked SurfaceViews this seems not to be the case. SurfaceViews render their content with OpenGL[22]. As we cannot access the Google Maps code this prevents stacked layers and results in merged content, i.e. one can hardly see the map area because it is placed over the panorama SurfaceView (Figure 4.17b). The map kind of "shines" through because the panorama renderer works with opacity mask to create a better 3D impression (see [47] or Section 4.1). It took some time to discover that the API provides the function *zOrderOnTop(true)* which has to be called on the GoogleMapOptions[23]-Builder on instantiation of the Google Maps fragment. When *zOrderOnTop* is set to *true* the Google Maps module is lifted over all other view elements (Figure 4.17c). Unfortunately this makes rendering buttons or similar interaction elements above the map area impossible as such elements would always be hidden behind the map. A theoretical solution would be to put those elements, e.g. a button to center the map at the user location, into an additional map overlay similar to the *LocationOverlay*. In order to keep the elements fixed (normally they would be moved when the map is panned) their coordinates would have to be updated on each camera update on the map.



(a) Illustration of overlaid contents. (b) Map "shines" through the panorama's alpha mask. (c) Screenshot of correct z-ordering.

Figure 4.17.: Z-ordering considerations

4.4. Miscellaneous Extensions

The following section lists some miscellaneous extensions made to the application. This includes the simplification of the server-client communication, the development of a CSV-based path editor and a builder-pattern implementation of the *Log* module.

[22]`http://www.opengl.org/` (last accessed: 04/01/2013)

[23]`http://developer.android.com/reference/com/google/android/gms/maps/GoogleMapOptions.html` (last accessed: 04/01/2013)

4.4.1. Server-Client Communication

In the initial implementation the client was represented by some loosely coupled threads communicating with the server application. Furthermore all the communication code was merged with the main logic code. This made making modifications to the server-client communication a straining process. Hence the logic for sending messages between server and client was put into an extra *ConnectionManager* module. This class manages the establishing and closing of a connection, the automatic sending of heartbeat messages to the corresponding counterpart to signal the own aliveness and especially it is responsible for exchanging the arbitrary location update messages. When such a message is received the *HallwayView* main module is notified over a Handler[24] it passed to the *ConnectionManager* on instantiation. The Handler code itself is split across several methods. Depending on the type of the received message it decides to handle it as a simple connection state message (connection established/closed/failed) or as a server message. A message sent by the server may be a simple heartbeat which does not require additional actions, a location update message which is handled by a separate *handleTextureMessage(...)* method or a *Finish* message which is sent at the end of a path and closes the currently active mode and forces the application to return to the start screen.

The interaction concepts of the manual mode make it necessary to enhance the server. The re-localization and manual panorama change request have to be answered automatically. Thus, such messages are identified and according to the requested action the corresponding location is searched in the path representation. Then the location information is automatically packaged into a response message and the answer is sent back to the client.

Especially when developing with an IDE like Eclipse[25] the separation of code into several methods and classes makes developing more efficient. This was also the reason why a project containing all shared code between server and client like *Path*, *SubPath* or *Message* was created. Eclipse provides the possibility to share a project as a library between other projects. So the server and client projects can access the same code and updates do not have to be copy-pasted across different class files.

4.4.2. Path Creator and Editor

Originally the path for the study in Soulard's thesis [47] was hard-coded into the server application code. Several FOR-loops iterated the location identifiers contained in the path and added them to the according *SubPath*s. As in this study at least three paths had to be created this would have resulted in many lines of code. Which would have been little extensible and changes to the path would have been difficult.

[24] `http://developer.android.com/reference/android/os/Handler.html` (last accessed: 04/02/2013)
[25] `http://www.eclipse.org/` (last accessed: 04/02/2013)

The new implementation is based on a CSV file denoting the *Location* identifiers, the orientation and the angle of a *SubPath*. Thus each line of the file represents a *SubPath* and consists of three parts: the first describes the included *Locations*, the second tells about the orientation of the *SubPath* and the third gives the angle to the next *SubPath*. The parts are separated by a semicolon, the locations by a comma. As sometimes a *SubPath* proceeds against the walking direction of the recording trolley the *Location* identifier ranges may also be specified in reverse order. Table 4.1 gives an example of a path description in the CSV format.

Table 4.1.: Example path representation in CSV format.

line	data*	locations*	orientation	turn
1	`003-039,085,060;;90`	003,004,. . . ,039,085,060	**	right
2	`102-124;;0`	102,103,. . . ,124	**	ahead
3	`146-125;120;-90`	146,145,. . . ,125	120°	left

** these are not IDs from the NAVVIS dataset*

*** the orientation is calculated automatically if no value is given*

This file structure is interpreted by the *PathCreator* and the necessary location panorama images are read from the original NAVVIS dataset. Those images are automatically copied, renamed and resized. The information about the location coordinates is automatically parsed from the NAVVIS information database and concatenated in a minimized CSV file. Thus the image data and the information data is kept at a necessary minimum. In order to make the created path accessible to the server and client applications the data has to be copied to the SD cards of the devices. The panorama images are stored in `/sdcard/[phoneserver|hallwayview]/pathXXX` folders where `XXX` is the three digit path number with leading zeros. The server application additionally needs the location information file which is put into `/sdcard/phoneserver/settings/pathXXX_info.csv` and of course the path information file which is put into `/sdcard/phoneserver/settings/pathXXX.csv`. The server includes a module to parse these files in order to build the corresponding path hierarchy.

4.4.3. Enhanced Log Module

All kind of data is logged by the client application. Again the CSV format is used to save the log. This is necessary to be able to understand and analyze users' behavior. Therefore the given *Log* module provides a variety of optional parameters which all have to be passed to the constructor. This makes creating log data tedious and space consuming, i.e. the code to define log data is very long. Either many overloaded methods have to be provided or many default values have to be passed to the log creating methods. The enhanced *Log* module implemented during this thesis follows the builder pattern, a software pattern especially designed for such a case. The proper *LogData* has no public constructor. So it has to be instantiated via the *LogData.Builder.build()*

method. This statement would create a *LogData* object with minimal but obligatory data. All further options may be passed to the provided builder methods. See the following code example (Listing 1) which compares the old and the new version of logging data. Notice the possibility for "fluid" programming as the builder methods return a pointer to the builder object.

```
/* old style method with a lot of default values given */
function log(eventType, now, now - mInitTime, null, null, "", -1, mAppMode, ...){ ... }
/* old style alternative: overloading method which calls the method above */
function log(eventType, now, now - mInitTime, mAppMode, mPathNumber, currentPreviewId){
        log(eventType, now, now - mInitTime, null, null, "", -1, mAppMode, ...)
}
/* Builder pattern: first set obligatory fields then provide optional parameters */
LogData d = new LogData.Builder(eventType, now, now - mInitTime)
        .evalMode(mAppMode)
        .path(mPathNumber)
        .messageType(mType)
        .previewId(currentPreviewId)
        .build();
log(d);
```

Listing 1: Comparison of logging codes.

The data contains for each log event the absolute (A) and relative (B) time in milliseconds. The latter is relative to the start of a specific presentation mode. It follows the event (C) and message (D) type preceded by the ID (E) of the current location and the ID of the displayed location (F). Column G holds the information about the path identifier. The metric of the distance information (H) and the orientation mode (I) are irrelevant for this thesis. The instruction mode (J) is logged in the last but one column. The last information (K) tells whether the message was sent by the experimenter or sent automatically by the server logic. A detailed summary of columns and possible values can be found in Appendix F.

Table 4.2.: Log data example in CSV format for the manual mode (J:2) on path A (G:1). Every log starts with an initial message (C=3,D=4). It follows the first location entry sent by the experimenter. Some time later an instruction request event by the user (C:4) is logged. The automatic server response follows immediately. The sixth row shows a touch event on the panorama view (C:7). Each log ends with a finish message (C:2).

A	B	C	D	E	F	G	H	I	J	K
1363452251986	0	3	4	-1	0	-1	Meters	1	2	FALSE
1363452262005	10019	0	2	100017	100003	1	Meters	1	2	TRUE
...										
1363452392757	140771	4	-1	200585	200583	1	Meters	1	2	FALSE
1363452393062	141076	0	2	200585	200585	1	Meters	1	2	FALSE
1363452402155	150169	7	0	200595	200585	-1	Meters	1	2	FALSE
...										
1363452496889	244903	2	-1	200043	200043	1	Meters	1	2	FALSE

Chapter 5.

Initial Study

The following chapter contains a description of the first user study. The research questions, the study settings and the results are presented.

5.1. Research Questions

The first survey was carried out with the aim to answer the research questions below:

RQ1. Which mode is preferred by users?

Depending on the situation each of the three modes has its advantages and disadvantages. While the fully automatic mode can stress users with too many panorama updates, the decision-point mode may confuse users by showing too little information. With the manual mode users gain the possibility to decide on their own when to change panoramas. In theory this can reduce visual and cognitive load at the risk of additional disorientation.

RQ2. Which mode assists users best in terms of time needed per path?

The assumption that the different modes with their different kinds of presentation and interaction possibilities influence the walking speed and the time needed for decision making of users led to RQ2. Based on qualitative data it is analyzed which of the three modes assists the user best, i.e. offers the fastest guidance to the goal.

RQ3. Is there a difference between the fully automatic and the decision-point mode?

This question evolved from RQ2. In more detail the differences between the two automatic modes are investigated. Is displaying decision-point panoramas automatically sufficient or is the user relying on the constant panorama updates? Answering this question is especially important in relation to indoor localization. A system displaying only decision-points does not need to know where the user is exactly. It would be sufficient to know whether he or she is at the beginning or the end of a hallway.

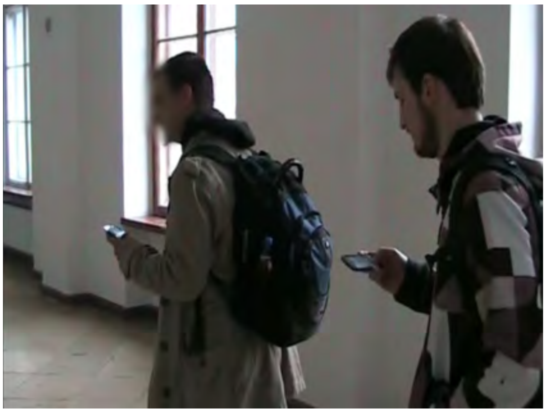
RQ4. What is the preview behavior of users in the manual mode?

Before being able to provide further tools that help users to orientate inside buildings it has to be investigated how users behave when given the possibility to switch through the panoramas on their own. Based on their actual location, are they viewing past or future panoramas? The study examines usage patterns and analyzes the switching behavior.

5.2. Setup

Conducting this initial study and answering the previously listed research questions serves the goal of gaining further insights. Before entering the next iteration of development user feedback was collected to be able to adapt the concepts mentioned above (see Section 3) to users' needs.

The application was evaluated in a *wizard-of-oz* [46] manner. The subjects receive a mobile phone running a client application pretending a server connection. In fact it is connected to another mobile phone managed by the experimenter. The location updates and path information are sent manually from this administration application (Figure 5.1). This setting and the interaction between the two applications is described in full detail by Soulard [47] and Möller et al. [45].



(a)

(b)

Figure 5.1.: The experimenter walking behind the participant (a) and the *wizard-of-oz* managed localization (b).

Before starting their run the subjects are briefed according to a standardized information sheet (Appendix A), they are given the possibility to initially test the application and eventual questions are answered. Each subject uses the application in each of the three modes on three different paths. The three modes are encoded in the following conditions:

$C1$ fully automatic mode

$C2$ decision-point mode

$C3$ manual mode

To prevent learning effects or at least to keep them at a minimum the order and the combination of paths and conditions are switched according to the pattern in Table 5.1. So subjects one to three start on path A, subjects four to six on path B and subjects seven through nine start on path C and so forth. Those starting on the same path start either with $C1$, $C2$ or $C3$ but never with the same condition. Instantly before conducting the study on a certain path-mode combination the subjects are briefed again to make sure what mode they will be using. The initial direction is explained but users are never told where the path will end. Neither they are told if the goal is on the same level (i.e. if they have to use stairs or elevators) nor if it is in the same building. The only information they get is that the path will end in a hall or hallway and that the goal is not in an office or lecture room.

Table 5.1.: Path and condition combinations for the first study.

	1st run	2nd run	3rd run
1	C1,A	C2,B	C3,C
2	C2,A	C3,B	C1,C
3	C3,A	C1,B	C2,C
4	C1,B	C2,C	C3,A
5	C2,B	C3,C	C1,A
6	C3,B	C1,C	C2,A
7	C1,C	C2,A	C3,B
8	C2,C	C3,A	C1,B
9	C3,C	C1,A	C2,B
10	C1,A	C2,B	C3,C
11	C2,A	C3,B	C1,C
12	C3,A	C1,B	C2,C

The three paths are located in the city campus building of the TUM[1]. While choosing the paths successions of hallways were selected in order to create equally "difficult" paths. In general the "difficulty" of paths is given by the number of turns included. Hence the total length, the number and the succession of turns and turn possibilities are chosen in a way that shorter paths got more

[1]Arcisstraße 21, 80333 Munich

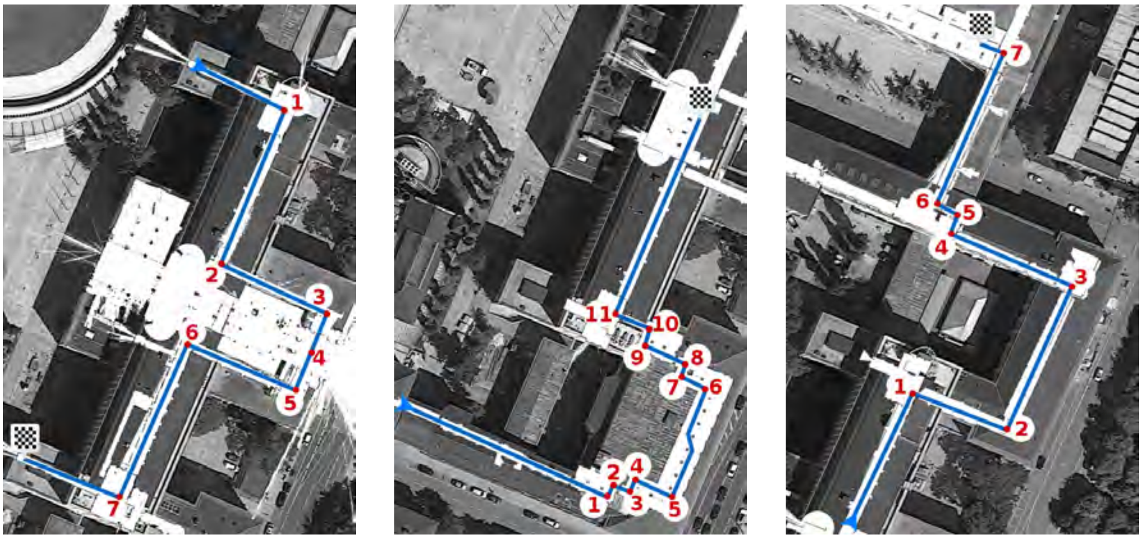
turns and longer paths vice versa. The resulting three paths are displayed in Figure 5.2. Path A is the longest path with 333 meters length. There are six turns on the way to the goal. In Figure 5.2a seven red dots mark the decision-points of path A. That is because the panorama images of A come from two different datasets and an additional decision-point (4) was needed to connect the two datasets to one single path. At this point users simply received a "go ahead" instruction. The second decision-point on path A is a fork from the main hallway which leads to a short part outside the building. The shortest path is B. It has a total length of 220 meters and contains 11 decision-points. There are two sections with turns followed directly by another turn. This zigzag makes it more difficult than the other two paths because the user gets more instructions and has more possibilities to take the wrong way. Path C is a combination of A and B. With its 316 meters it is not as long as path A and not as winding as path C (only seven turns). Equally to path A it contains forks from the main hallway and similar to C there is a turn after turn combination.

If subjects decide wrong and take an incorrect path they are told that they are wrong by the experimenter. Additionally they are encouraged to look at the panorama again. If this does not help them either they receive a hint which door, hallway or wall they are looking at on the panorama picture and where it is in their real environment. Participants being indecisive about the next decision get similar help after some time. Mostly it was sufficient to call their attention to some eye-catching objects like fire extinguishers, waste bins or exhibition showcases. If this did not convince them that the displayed panorama reflected their environment the experimenter helped them out with a hint to go in a particular direction.

During the run the subjects are encouraged to give feedback in a *Think aloud* manner [49]. Afterwards they are asked to answer a questionnaire which provides the possibility to note down this feedback, too (Appendix B).

5.3. Participants

Participants are students, colleagues and friends. 12 subjects take place in the initial study. 75% are male and 25% are female. The average age is 26 years with a standard deviation of 3.2. Asked if they own a smart phone 75% answered "yes". Only one participant has experience with indoor navigation systems, i.e. 91.6% did not have any experience with this kind of navigation before the study.



(a) Path A (b) Path B (c) Path C

Figure 5.2.: Paths in study A with start/end markers and highlighted decision-points.

5.4. Results

In this section the quantitative and qualitative results of the survey are discussed. The quantitative data is gathered by questions in the questionnaire. The questions and statements can be answered using 5-step Likert-scales reaching from "I strongly disagree" to "I strongly agree". The qualitative findings are based on automatically collected data. The prototype logs user interaction in combination with time and location data. Time measurements on how long users needed to pass the paths are calculated by subtracting the time of the first log entry from the last log entry's time. The automatic logging is started as soon as the participants receive the first instruction and it is stopped after they reach the destination. The calculations and findings based on the gathered questionnaire and log data are presented in this section.

5.4.1. RQ1. Which mode is preferred by users?

The first two questions of the quantitative analysis are *"I found the method pleasing to use"* and *"I felt guided well to the goal"*. When asked if the fully automatic mode was pleasant to use most of the users strongly agree ($1.66, SD = 0.47$). This is displayed in Figure 5.3 in a box plot without whiskers with all data between *"I agree"* or *"I strongly agree"*. Contrarily the decision-point and manual modes are rated neutrally with respect to how the users are pleased with them. In detail the decision-point mode reaches an average degree of approval of 0 (neutral, $SD = 1.00$) just as much as the manual mode with a standard deviation of 0.91.

That the fully automatic mode is the user's favorite mode gets even clearer when considering the evaluation of the question how well the users feel guided to the goal by the specific mode. All subjects either agree or strongly agree. This results in median value of 2.0. In fact 75% answer the question *"I felt guided well to the goal"* with *"I strongly agree"* $(1.75, SD = 0.43)$. Using the other two modes users feel guided well to the goal, too, but not as well as with the fully automatic mode. Again these modes gain neutral results with a positive tendency: decision-point $(0.67, SD = 0.94)$, manual mode $(0.42, SD = 0.76)$.

Running a Mann-Whitney's U test to evaluate the differences in the responses of the 5-step Likert scale questions reveals a significant effect of group. It is proved that in terms of pleasantness the fully automatic mode outperforms the decision-point $(median_{C1} = 2.0, median_{C2} = 0.0, W = 130, Z = 3.51, p \ll 0.05)$ and the manual mode $(median_{C1} = 2.0, median_{C3} = 0.0, W = 136, Z = 3.86, p \ll 0.05)$. Moreover, guidance abilities of the fully automatic mode are rated significantly better than in the other two modes. Following Mann-Whitney's U tests the fully automatic mode differs significantly from the decision-point mode $(median_{C1} = 2.0, median_{C2} = 0.5, W = 117, Z = 2.82, p < 0.05)$ as well as from the manual mode $(median_{C1} = 2.0, median_{C3} = 1.0, W = 133.5, Z = 3.80, p \ll 0.05)$. Using the same test to evaluate the differences between the decision-point and manual mode gives that both mode neither differ significantly in terms of pleasantness nor in terms of guidance.

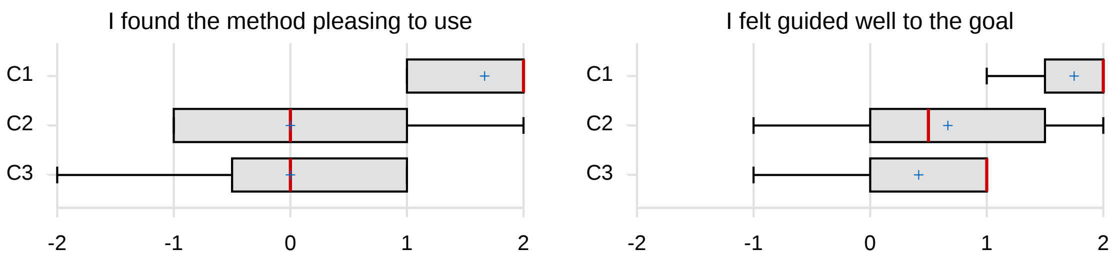
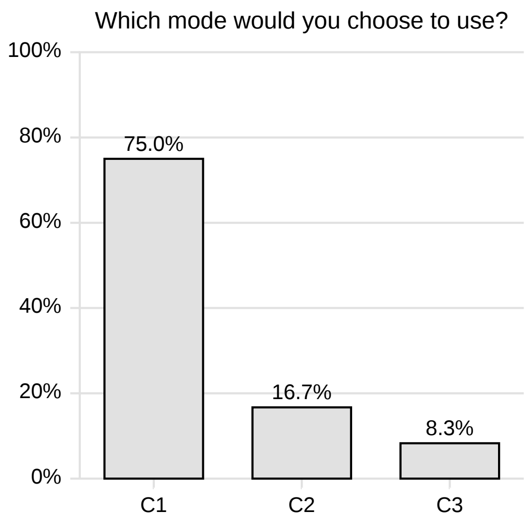


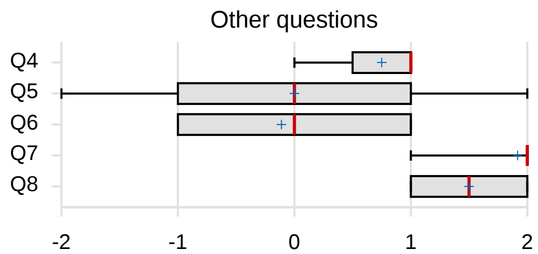
Figure 5.3.: Boxplot analysis showing the subject's preferences (the red bar represents the median and the blue cross denotes the mean value).

As one can see in Figure 5.4a, 75% of the users would like to use the fully automatic mode if they could choose between the three modes. While 16.7% would like to be guided by the decision-point mode only 8.3% of the users would like to change the panoramas manually. This result shows once more that the users strongly prefer the fully automatic mode but dislike the manual mode. This gets even clearer when considering the response to the the question *"Changing the panoramas in the list view is useful"* (see Q5 and Q6 in Figure 5.4b). The possibility to select the panoramas manually from the list view was rated negative $(-0.11, SD = 0.87)$. Equally not all subjects considered changing panoramas by swiping up and down as useful $(0, SD = 1.13)$.

Summary The analysis shows that the subjects prefer the fully automatic mode ($C1$). Getting panorama updates every few meters they feel safest and best guided to their goal. They are most pleased with the application when using it in the fully automatic mode. One participant summed it up with *"The fully automatic mode worked flawlessly, the instructions were clear and distinct, getting lost was nearly impossible"*. Another one stated that the manual mode would be better suited if one already knows the building or environment.



(a) 9 of 12 subjects would choose the fully automatic mode.



(b) Boxplots of questions 4 to 8 in the questionnaire.

Q4 Panoramas of decision-points are sufficient for orientating

Q5 Changing panoramas by swiping up and down is useful*

Q6 Changing panoramas in the list view is useful**

Q7 The ability to re-localize myself is useful*

Q8 The method of moving up the phone to re-localize is convenient*

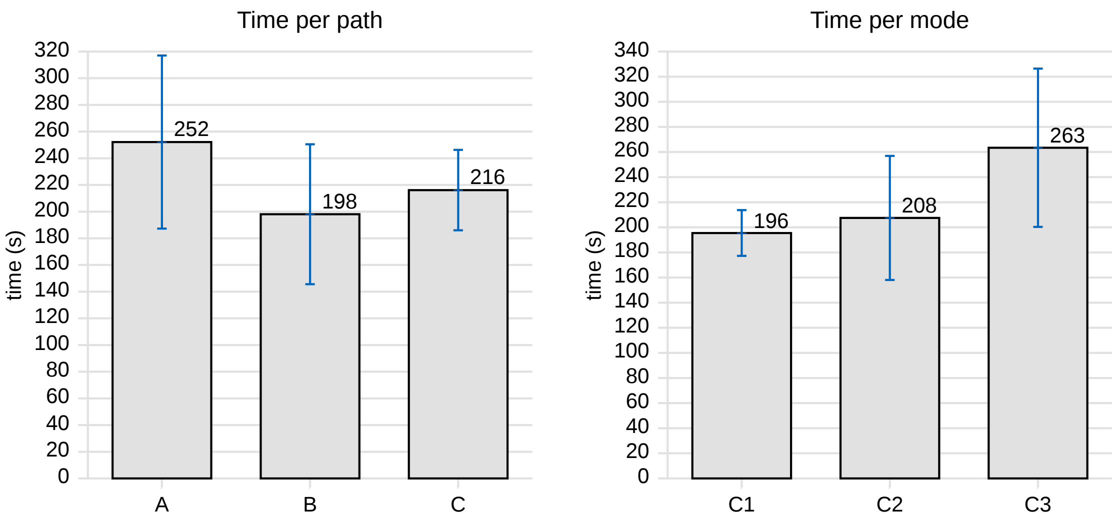
* One participant did not answer this question

** Three participants did not answer this question

Figure 5.4.: Graphical analysis of questionnaire entries Q3 (a) and Q4-Q8 (b).

5.4.2. RQ2. Which mode assists users best in terms of time needed per path?

The response to RQ2 is based on the evaluation of the qualitative log data. Time is measured based on the automatic log data from the moment the user start the run until the last location is reached. Figure 5.5a shows the average time per path. The time needed by users to reach the end of the path corresponds with the path lengths. In average users need 252 seconds on path A ($SD = 64.8$), which is the longest path with 333 meters. Passing path B (shortest path, $220m$, $SD = 52.4$) takes averagely 198 seconds. Path C, which has a distance of 316 meters, is passed in a mean of 216 seconds ($SD = 30.1$). As expected the length of the paths influences the walking time. The time analysis suggests that no path is more difficult than another or disturbs the time measuring in any not foreseen way.



(a) Time analysis per path.

(b) Time analysis per mode.

Figure 5.5.: Average time per path and mode.

With the modes being the independent variables of the experiment it is interesting to evaluate their influence on the dependent time variable. Figure 5.5b illustrates this in detail. It gets obvious that subjects averagely need longer to get to the end of a path when using the manual mode ($263s, SD = 63.1$). Using the manual mode takes longer because often the users did not know exactly which locations they had already passed and which not. So they changed the panoramas forward and backward until they were sure again where they were. Of course this takes additional time. Another point why users are slower when using the manual mode are the additional interaction possibilities (Section 4.2.3). Users were introduced to all functionalities of the manual mode in advance to the task however most users tested the different swipe and double-tap gestures, opened the path preview and explored the functions of the panorama gallery once they had started their task. This means, they stopped and interacted with the device unnecessarily. However, both automatic modes are about one minute faster than the manual mode. With the fully automatic mode ($C1$) it takes about 196 seconds ($SD = 18.2$) to reach the goal while with the decision-point mode participants need averagely 208 seconds with a standard deviation of 49.5. Compared to each of the modes $C1$ and $C2$ the manual mode result is significantly different (for both $p \ll 0.05$ in a Student's T test).

Summary In terms of time needed per path the fully automatic mode ($C1$) guides the user best. The log data states that it is the fastest of the three modes. In contrary using the manual mode is the slowest alternative. I.e. providing the user with automatic updates offers better guidance than giving users the opportunity to manually update the route information. The next paragraph analyzes whether it makes a difference to use the fully automatic or the decision-point mode.

5.4.3. RQ3. Is there a difference between the fully automatic and the decision-point mode?

The observation of Figure 5.5b shows that condition $C1$ and $C2$ resulted in quite the same time performance. A Student's T test gives that this difference between the both automatic modes is not significant ($p > 0.05$). This leads to the assumption that it does not matter if the user is provided with continuous panorama updates or decision-point panoramas only. This statement is further confirmed by the quantitative data from the questionnaire. Question Q4 states that users, despite they prefer the fully automatic mode, think that panoramas of decision-points are sufficient for orientating $(0.75, SD = 0.43)$.

Summary The both automatic modes differ not significantly from each other. Thus it is assumed that panoramas of decision-points are sufficient for orientation as it makes no difference in time if the user was either guided by the fully automatic mode displaying all panoramas or the decision-point mode displaying only panoramas of turn locations.

5.4.4. RQ4. What is the preview behavior of users in the manual mode?

By analyzing the number of manual panorama changes (via the up or down swipe gesture) and re-localizations per path it is found that users change the panoramas manually about 15 times $(15.25, SD = 4.55)$. There are two participants needing more than 20 swipes but the rest ranges between 10 and 18 manual changes. The evaluation of the usage of the re-localization function gives that there are two groups. One uses the functionality a lot (> 16 uses) while the other group rarely uses it (< 3 uses). This leads to the conclusion that, once users discovered how powerful and comfortable the re-localization method is they use it to get constant panorama and instruction updates like in the fully automatic mode. Users with a good sense of direction and those who did not get lost or insecure about their current location did not use the re-localization functionality intensively. There is no causality between the use of the swipe gesture and the re-localization method. Table 5.2 shows that there are users with many swipes but little re-localizations (participant 8), users with a high number of both (participant 3) and users with some swipes and many re-localization actions (participants 1,2,5 and 7). Neither, it can be found a dependency between the number of interactions and the kind of path.

In average participants used the re-localization for the first time after 132.8 seconds. This corresponds to 49.8% of the average path duration and 33.9% of the average path length, i.e. the first use of the re-localization method is after one third of the way but after half of the time. This

[2] The first re-localization event here was logged within the first five seconds (unlikely to be intended) so we took the next event

[3] Participant 6 did not use the re-localization at all

Table 5.2.: Number of swipe gestures and re-localizations per subject.

	swipe	re-localizations (1st after . . . seconds)	path
1	11	18 (70)	3
2	13	24 (49)	2
3	23	20 (149)	1
4	12	1 (35)	1
5	15	16 (1/145)[2]	3
6	11	0 (0)[3]	2
7	10	20 (84)	2
8	25	3 (104)	1
9	16	2 (231)	3
10	16	2 (176)	3
11	18	2 (99)	2
12	13	3 (3/319)[2]	1

leads to the assumption that users are faster after they used the re-localization as they cover two thirds of the distance in one half of the time from then on. Of course we keep in mind that with the time the subjects get used to the system and that this familiarization has an impact on the walking speed, too.

Additionally an analysis of the distance between the actual user location and the previewed panorama was conducted. When subjects select a future panorama the distance is positive whereas when they are looking at already passed location panoramas it is negative. According to this analysis users look at future panoramas in 47.5% of time. The percentage of time looking at upcoming panoramas is 25.4% for the group of users using the re-localization functionality often (with more than three re-localizations). Those using it rarely, preview panoramas to come in 63.3% of time. Again assuming that the latter group consists of people with a good sense of direction we deduce that those subjects look at the selected panorama, compare it to e.g. the end of the hallway, recognize the similarity and then walk on to this location. As they reach it they swipe to get the next panorama and move on with the former mentioned behavior. The "many-usage" group however is not as self-confident and relies more on the correctness of the system. This implies some waiting until the next panorama is loaded and some time until the user matches it to the actual environment. As a result the "little-usage" group is about 6.8% faster.

Figure 5.6 depicts a time based plot of user interactions. The blue graph stands for the distance between the actual position and the location of the currently preview panorama. The red graph denotes the position relative to the path and helps interpreting users' movements. The violet and green vertical bars show the swipe and re-localization events. At the beginning subject 4 walks on seeing the initial picture. Therefore the distance gets negative. After the first swipe and a movement towards this location the subject seems to be irritated as there are forward changes

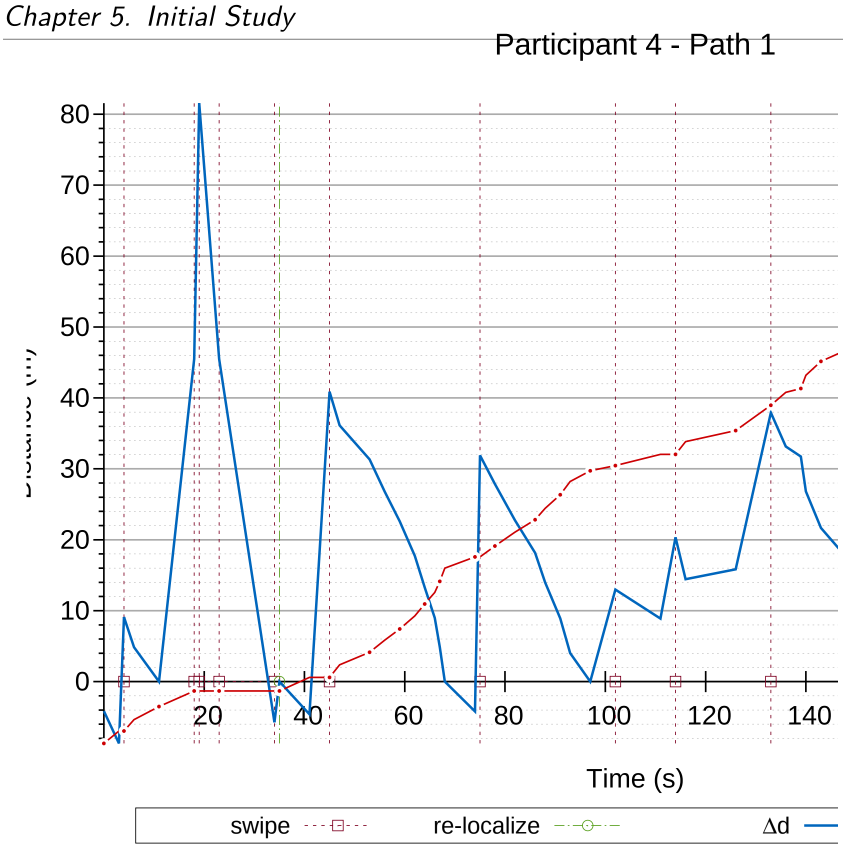
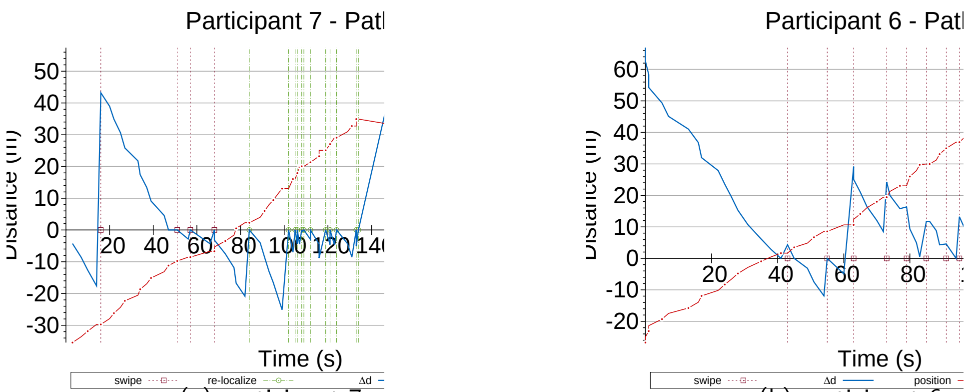


Figure 5.6.: Distance and position plot for participant 4 on path A.

followed by backward changes (distance raises and falls again while the user's position stays the same). The irritation about the actual position is cured by a re-localization. Then the subject shows the former mentioned behavior of swiping, matching and walking.

As already mentioned before there are no usage patterns that can be associated to a specific kind of path. Figure 5.7 shows how distinct subjects use the manual mode on e.g. path B. Participant 7, let us call him Bob from now, fits good into the group of heavy re-localization users. First he begins using the swipe gesture to switch the decision-point panoramas. He performs well, manages the first zigzag without problems ($60s$) and then he discovers the re-localization method. Whether he really needs it at this point is not know. As Bob realizes how the re-localization works he uses it to get continuous location information. This usage pattern guides him well through the second zigzag on path B. Before entering the final hallway Bob tries the swipe functionality again ($150s$). The system now shows him the final panorama. As the hallway is quite long and Bob cannot see the goal yet he gets irritated. Still viewing the final panorama he even walks back to compare it to a location he had already passed. With the help of the re-localization method Bob returns to the final hallway. Again he wants to use the swipe method to finish his task ($180s$). Obviously it cannot help him because he cannot map the displayed panorama of the goal to the environment at the end of the hallway. After repeated backward and forward swipes Bob walks to the end using the re-localization method.



(a) participant 7 (b) participant 6

Figure 5.7.: Distance and position plot for participant 6 and 7 on path B.

The predecessor of Bob, let's call her Alice, is a good example for the "little-usage" group as she never uses the re-localization method. The plot shows a positive distance at the very beginning as Alice seems to have swiped to the first decision-point before the first log event. She walks along the path using the swipe-compare-walk iteration pattern. Having reached the last-but-one decision-point ($95s$) she swipes to see the instruction at the last decision-point. Even before she reaches the last turn she matches the seen panorama to the location she sees in front of her and switches to the panorama of the goal. Without hesitating she turns right into the final hallway and walks towards the finish. At about the same location where Bob repeatedly swiped back and forth (Figure 5.7a, $180s$) Alice swipes back to the last instruction to ensure that she is right. Some steps ahead she swipes again to the goal panorama and reaches the goal 70 seconds earlier than Bob.

Summary It is observed that there are two different behavior patterns for subjects using the manual mode. These patterns occur on all paths and are not dependent on distinct mode-path combinations. The first pattern is applicable to a group of users who use the manual mode in combination with the re-localization method to get ongoing panorama and instruction updates. In other words, they use the manual mode like the automatic one. The second pattern manifests with users with a good sense of direction who orientate with the help of some few panoramas. As they are less dependent on the responses of the system they reach the end of a path faster. For people with a good sense of orientation or those who already know a building vaguely the manual mode seems to be an alternative to the automatic modes. One participant quoted *"I suppose the manual mode is suited better if one already knows the environment"*. The example of Bob illustrates how difficult it seems to be for users to orientate again on their own when they once gave the responsibility to an automated system.

Unfortunately the data does not allow to make conclusion about the maximum distance between two decision-points. Obviously users have problems to match preview panoramas to real locations that are far away. Investigating a threshold for the maximum distance between two decision-points would be helpful. Another interesting point is how many future decision-points users can skip by manual changing the preview until a re-localization or other kind of help is needed.

5.5. Discussion

The goal of this initial study is to evaluate differences between automatic modes managed by a computer assisted system and a manual mode controlled by the user. Furthermore the study aims at analyzing powers and weaknesses of a fully automatic mode compared to a decision-point mode. The former offers way instructions and indoor panoramas every few meters while the latter only provides panoramas and instructions at turn locations.

Asking users which method is most pleasing and which offers best guidance to the goal leads to a superior rating of the fully automatic mode. The findings indicate as well that the manual mode is liked least by users. Moreover, getting way instructions automatically lets users reach the destination faster. Analyzing the time per path shows that sifting manually through the instructions significantly enlarges the time needed to walk to the end of the path. In contrary, there is no significant difference between the fully automatic and the decision-point mode. This leads to the assumption that showing instructions at decision-points is sufficient. This hypothesis is also confirmed by the responses of users to the question if panoramas (and instructions) at decision-points are sufficient for orientating.

At this point it should be mentioned that the *wizard-of-oz* setting used in this study guarantees an always accurate localization of the user. In case of a less exact localization the fully automatic mode will display wrong panoramas more often which can have negative effects on the preferences concerning this distinct mode. Contrarily the decision-point mode is less error-prone in terms of inaccurate positioning. Furthermore, the route information, indicating the distance to the next turn respectively the goal, of the decision-point mode contained an error. The distance was not updated while approaching the next turn, instead it always indicated a distance of 0 meters. This bug has to be fixed and the both automatic modes will be evaluated again in a second study.

In general, the manual mode offering the possibility to jump virtually to the following or passed decision-points irritates users. Users have difficulties to distinguish between panoramas belonging to locations they had already passed and panoramas referring to future locations. In fact, the manual mode offered a feature to mark locations as "already seen" but users did not use it. Moreover, the "already seen" tags can only be observed in a little prominent list preview of all decision-points. A future prototype could show an icon in the main panorama view to provide feedback about already passed location at a more popular place (see Figure 5.8a for a possible solution). One could also think about a combination of the manual mode and some kind of localization logic that marks already passed locations automatically. Especially for buildings with many identical looking hallways (e.g. hospitals) such a combined system would be advantageous as the matching of virtual panoramas to already passed locations gets nearly impossible in such situations.

(a) Panoramas tagged as "already seen" at a more prominent place.

(b) Optional replacement for left-right-left instruction combinations.

(c) Sketch for providing multiple instructions in a split view.

Figure 5.8.: Illustration of possible improvements identified in study A.

A kind of such an automatic positioning is already included in the manual mode. Users can trigger a re-localization which then shows them the panorama of their current location and the corresponding way instruction. This feature is especially useful in cases when users feel insecure about the future way or need more information about the correct path at their actual position. Indeed, seven users noted this feature when asked what aspects of the system they like most. This supposes to include the re-localization method in the other modes as well.

When subsequently used to get ongoing updates about the own position the re-localizing feature transforms the manual mode to be similar to the fully automatic mode. This and other interaction behaviors are found in a usage pattern analysis. The evaluation considering especially the locations and numbers of interactions with the manual mode reveals two main user groups. While participants of the first group use the manual mode's features frequently those belonging to the second group use them rarely. A special kind of plot which puts panorama change and re-localize events in relation with the users position on the path facilitates a graphical per user analysis. The resulting findings give rise to the assumption that subjects of the second, "little usage" group have a good sense of direction. This is why they have an increased confidence about their actual position, it makes comparisons between virtual panoramas and real environments easier and thus leads to less use of aids provided by the system. The first group seems to be a set of users with less abilities to orientate. Hence, they get "lost" more often while switching between the virtual panoramas and consequently rely on the localization system and try to get information about their current position by using the re-localization feature.

In addition, the initial study reveals a certain dissatisfaction of users with the presentation of panoramas and instructions. Especially fast instruction changes, like in the left-right-left combinations of path B, require other techniques than the present arrow displaying only 90° turns. One could imagine an arrow formed like an "S" or a split screen displaying the next two decisions

(a) In this situation one participant wanted to enter the bathrooms behind that door because the direction arrows is positioned badly.

(b) An underexposed panorama making it difficult to identify reference points.

Figure 5.9.: Examples of locations in the path definitions of study A which give rise to improvements.

at the same time. Sketches in Figure 5.8 illustrate these ideas. Another point which has to be worked over in future iterations are the individual path definitions. Some turns in the current path representations are located badly. Either the images at those locations are underexposed which makes matching the panoramas to the real environment more difficult or the direction arrow leads to wrong decisions. Examples are shown in Figure 5.9.

Users of the decision-point and manual mode complained about the fact that adjacent decision-points are at times too far apart from each other. In such situations long and eventually dark hallways or obstacles like closed doors make the matching between virtual panoramas and real environments impossible. It makes no difference if the obstacles exist on the panorama images or in the real environment. The point is that if one can not anticipate that the displayed panorama is at the end of a hallway, this hallway will not be judged as the correct way. In a future study these situations have to be identified and intermediate decision-points have to be inserted in the path definitions in order to provide more reference points.

Chapter 6.

Follow-up study

The second user study aims at evaluating new system elements like the *MapPreview* (Section 4.3). Again the different modes are evaluated in terms of user satisfaction, usability and time needed per path. Additionally insights on locations of decision-points and importance of way instructions should be gained. This chapter presents the research questions, settings and findings for the second study.

6.1. Research Questions

RQ1. How do users rate the enhanced instruction modes?

This question is concerned with assessing which mode is rated best in terms of pleasure and guidance. Especially the enhancement and improvements made to the mode presenting decision-points only motivate an examination of the relation between this mode and the fully automatic one. A special emphasis is put to the relation between the mode ratings from the first and second study.

RQ2. Which mode offers fastest guidance to the destination?

Depending on the time needed per path it is analyzed which mode assists users best, i.e. guides users to the goal fastest. Besides the evaluation results from the second study a comparison to the results from the first study is given.

RQ3. What is the relation between panorama and map usage?

This question aims at the ratio between panorama and map usage. On the one hand the subjective estimation of users about their map and panorama usage is analyzed based on the questionnaire responses. On the other hand a comparison between panorama and map interaction based on the qualitative log data is presented.

RQ4. What are important locations for way instructions?

This research question clarifies where users find instruction updates important and necessary. How far these locations match the existing decision-points is evaluated based on the data gathered by the manual mode. Additionally the answers related with instruction importance in the questionnaires are analyzed and the results are interpreted.

RQ5. What system elements do users consider as especially good or bad?

Are there redundant elements in the system that can be removed? Exist elements that users consider especially valuable? User feedback concerned with UI elements and system features as well as proposed improvements are examined in the answer to this research question.

6.2. Setup

All in all the study setup is the same as in the first study 5.2. The system is evaluated using a *wizard-of-oz* [46] setting. The subjects receive a mobile phone running the application. The experimenter controls the subjects' application from another mobile phone running an administrative application. In advance to the survey the participants are asked to read a standardized briefing sheet (see Appendix C) explaining the different modes and tasks. Each subject uses three modes encoded in the following conditions (note that the conditions in the second study are marked with an asterisk):

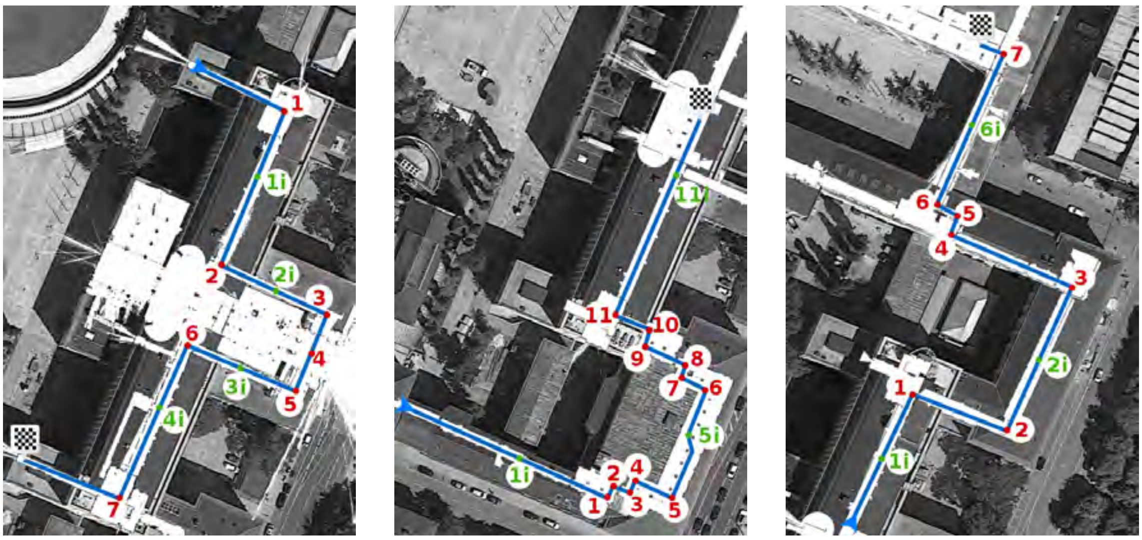
$C1^*$ fully automatic mode

$C2^*$ automatic decision-point mode

$C3^*$ manual mode

With $C1^*$ and $C2^*$ the task is simply to walk along the displayed route and to follow the provided way instructions. As explained in Section 4.2.3 the manual mode in the second study differs from the version used in the first study. The subjects' task with $C3^*$ is to request updated instructions and panoramas as soon as they feel insecure about the future path. Participants are motivated to keep the number of instruction requests at a minimum. To support this they are told that the destination is neither an office nor a lecture room to prevent requests at each door which might lead to the destination. In contrary to mode $C1^*$ and $C2^*$ (where the destination is displayed in the map preview) the participants are not told where they are walking to. Hence the destination is unknown some subjects' felt reminded of a "treasure hunt" or "paper chase" when using the manual mode.

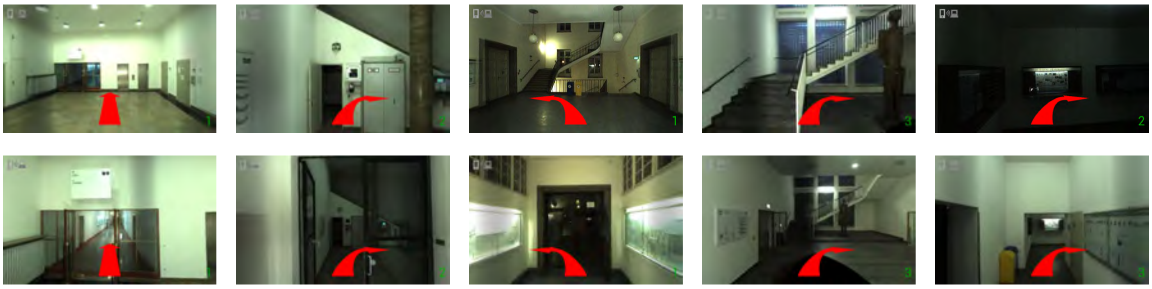
Equally to study A each user evaluate each mode on another path. The mode-path combinations were switched from user to user as defined in Section 5.2 in Table 5.1. The three paths from the first study are used again but single locations and their corresponding panoramas are updated. Wherever possible those panoramas misleading users in the initial survey are replaced (Figure 6.2). This affects mainly panoramas with very dark content or those located too near to path vertices what forces the direction arrow to point to unwanted doors, hallways or staircases. As mentioned in Section 5.5 users in the first study complained about "too far away" decision-points. Standing at the beginning of a long hallway they couldn't check if the displayed panorama, which shows the end of the hallway (the next decision-point) in mode $C2$, actually refers to the real environment at the end of the hallway. This leads to the inclusion of additional decision-points between those turns locations that are a great distance apart. These *intermediate* decision-points are marked green in Figure 6.1.



(a) Path A (b) Path B (c) Path C

Figure 6.1.: Paths in study B (start/end markers, red decision-points, green intermediate decision-points.

During the three tasks users are motivated to give immediate, oral feedback on the system similar to a *Think aloud* [49] setting. Afterwards all participants fill out a questionnaire in presence of the experimenter who answers potential questions and resolves uncertainties.



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

Figure 6.2.: Changed panoramas between the first (upper row) and the second study (lower row). Panoramas where the arrow is ambiguous (a), pointing to wrong door (b,c) or staircase (d) and dark and underexposed panoramas are updated.

6.3. Participants

The second study is conducted with 18 participants. 44% are students, 33% have an academic degree and the other 22% are workers or employees. 12 (67%) male and 6 (33%) female subjects took part in the study. The average age of the participants is 32.4 $(SD = 13.2)$ years. This is more than in the first study because 4 people older than 50 participated. 67% of the participants possess a private smartphone but do not use navigational applications often according to their answers on a 5-step Likert-scale from "never" to "frequently" $(0.17, SD = 1.41)$. Equally to study A only one participant had experience with indoor navigation before taking place in study B.

6.4. Results

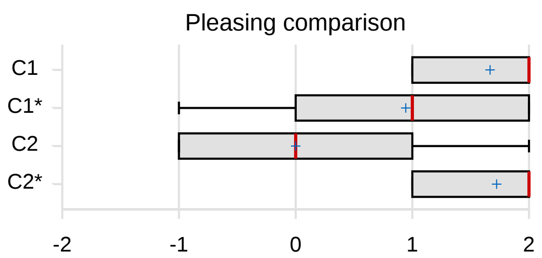
This section presents the results of the second study based on quantitative as well as on qualitative data. The quantitative data is mainly gathered from the answered questionnaires but as well from the oral feedback given by users during the tasks. Most items of the questionnaire could be answered using a 5-step Likert-scale. This section refers to statements requiring the agreement of users (strongly disagree, disagree, neutral, agree, strongly agree) unless specified differently. Finally, the questionnaire contains open-ended questions. Similar to the first study the qualitative findings are based on automatically logged data. The logging starts as soon as the user receives the first instruction and ends after reaching the goal when the experimenter remotely finishes the task. Some results are illustrated with box plots. The blue cross and the red line represent the mean and the median, respectively. The gray colored box encloses the lower and upper quartile and the whiskers sticking out of the box denote the maximum and minimum datum. In what follows labels like "Q7" refer to the seventh question in the questionnaire.

In contrary to the first study the manual mode is mainly used to track where users would find instruction updates useful respectively where they implicitly need new way instructions in order to be able to continue. That is why the manual mode is only considered by RQ4. The other sections leave out the manual mode ($C3^*$) and focus on the fully automatic ($C1^*$) and the decision-point ($C2^*$) conditions.

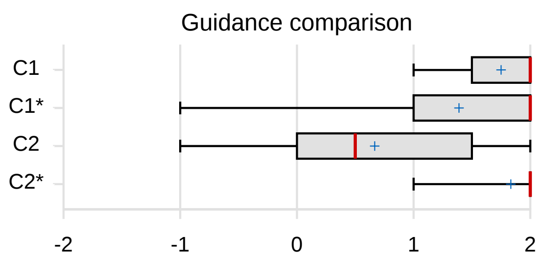
6.4.1. RQ1. How do users rate the enhanced instruction modes?

In order to find out how users are pleased with the distinct modes of study B they had to answer the statement "I found the method pleasing to use". Users agree that the fully automatic mode is pleasant to use ($0.94, SD = 1.0$) and they strongly agree on the pleasantness of the decision-point mode ($1.72, SD = 2.0$). Compared to study A these results show significant differences. According to a Mann-Whitney's U test the difference between the responses in study A and B concerning the fully automatic mode is significant ($median_A = 2.0, median_B = 1.0, W = 152, Z = 2.02, p < 0.05$). Similarly the increase of the medians for the decision-point mode from 0.0 in study A to 2.0 in study B is found to be significant ($W = 19, Z = -4.03, p \ll 0.05$). Figure 6.3a illustrates the difference between the ratings with box plots for conditions in the first and second study (marked with an asterisk).

In terms of guidance to the goal the fully automatic mode is rated with an average of 1.39 ($SD = 0.85$). With very little deviation users agree that with the decision-point mode they feel guided well to the goal ($1.83, SD = 0.38$) in study B (see $C2^*$ in Figure 6.3b). While the difference between responses related to the fully automatic mode do not differ significantly from the first to the second study, there is, according to a Mann-Whitney's U test, a significant effect leading to the increased rating of the decision-point mode in study B ($median_A = 0.5, median_B = 2.0, W = 36, Z = -3.47, p < 0.05$).



(a) "I found the method pleasing to use."



(b) "I felt guided well to the goal."

Figure 6.3.: Comparison of ratings for the first and second (*) study.

Summary The findings indicate that the enhancements made to the modes (see Section 4.2.4) make users favor the decision-point mode. This can be proved at least in terms of pleasantness.

A Mann-Whitney's U test run on the response distributions for $C1^*$ and $C2^*$ of the statement "I found the method pleasing to use" shows a significant difference between the two modes ($median_{C1^*} = 1.0, median_{C2^*} = 2.0, W = 90.5, Z = -2.52, p < 0.05$). Either the map view or the eliminated distance bug make the decision-point mode superior to the full automatic one. In contrary to the first study the decision-point mode provides additional route information in the second study. This seems to outperform the advantage of continuously updated panoramas and instructions in mode $C1^*$.

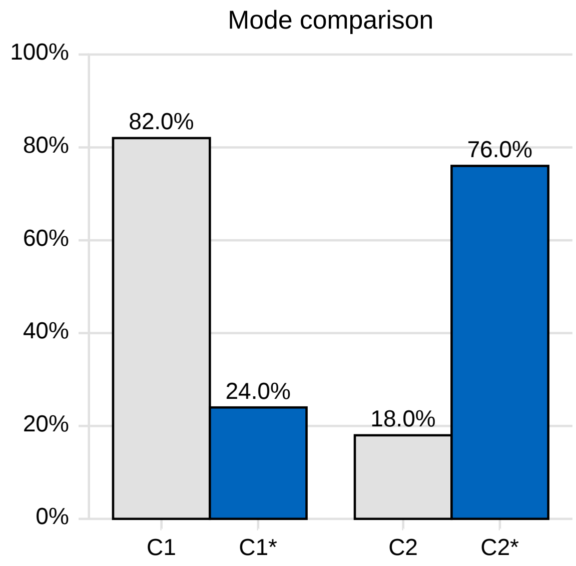


Figure 6.4.: Comparison which mode users prefer in study A (gray) and B (blue). Note that the manual mode ratings in study A are left out in this analysis.

Asked directly which mode they prefer only 24% of users choose the fully automatic mode. Considering only the both automatic modes in study A (i.e. leaving out the answers for $C3$) gives that 82% favored $C1$. This is a drop of 58 points from $C1$ to $C1^*$. The other way round the rating of $C2$ increases from 18% to 76% for $C2^*$. Figure 6.4 illustrates this contrast in a bar chart.

The hypothesis formulated in Section 5.4.3 that panoramas of decision-points are sufficient for orientating is confirmed. Users agree on the corresponding statement (Q5) in the questionnaire ($1.11, SD = 0.83$). This is no significant increase compared to the samples of the same statement in study A but it still confirms the agreement of users that decision-points are sufficient for orientating.

6.4.2. RQ2. Which mode offers fastest guidance to the destination?

Based on the first and last log entry for each participant the time per path and mode is calculated. The time per path ranges between 238 seconds on path A and 249 seconds on Path B. This is two seconds more than the average duration on path C. On the one hand the difference between the paths is marginal (11 seconds) but on the other hand passing the shortest path lasts longest. This is contradictory to study A where the time per path correlates with the path length but can be traced back to the fact that the way to the starting points of the paths was chosen differently in study B. It was avoided to lead subjects along future path segments, i.e. in study B those who did not know the building in advance[1], had no idea of the path. That means for example that in the second study the zigzag on path B represented a much more difficult obstacle which required more time to pass as decision making took longer.

In study B subjects using the decision-point mode are faster than those using the fully automatic mode. The average duration for $C2^*$ is 233 $(SD = 38.0)$ seconds while it takes averagely 239 $(SD = 38.0)$ seconds to reach the goal with $C1^*$. Similar to study A the manual mode is the slowest alternative with an average duration of 262 $(SD = 49.1)$ seconds. A Student's T test gives that the difference between the automatic and the decision-point mode is not significant. Only the comparison of the times of $C1^*$ and $C3^*$ and of $C2^*$ and $C3^*$ shows a significant difference ($p < 0.05$ for both).

Looking at the average durations per mode in the second study it gets clear that with all modes it takes longer, or at least the same time, to reach the end of the path than in the first study. The bar chart in Figure 6.5 illustrates this with gray boxes for study A and blue boxes for times resulting from study B. A Welch's T test on the unpaired samples of both studies reveals a significant difference between both versions of the fully automatic mode ($t(28) = 4.70,\ p \ll 0.05,\ Cohen's\ d = 1.59$) with a large effect size. The between-groups comparison of the results for the decision-point modes in study A and B states that the mean difference of 25 seconds is not significant.

Summary There are three main findings concerning guidance time in study B. First, the comparison of average times per path of study A and B indicates that additional route knowledge makes navigating through difficult route segments easier and faster. Second, no qualitative difference between the automatic and the decision-point mode can be found what further supports the hypothesis that continuously updated way instructions are not superior to instructions updated only at decision-points. Third, the additional information provided in study B slows down users on their way to the path destination. Averagely, in conditions $C1^*$ and $C2^*$ users are slower than in the same conditions in the first survey. At least for the fully automatic mode this difference is found to be statistically significant. One hypothesis is that users are over-strained by the amount of displayed information and need more time for interpreting it. Another one is that the additional information leads to a "100% save" mentality. Whereas a "trial and error" mentality may have led to faster decisions in the first study, the accurate map presentation may have encouraged users to

[1] 83% of the subjects stated not to know the building

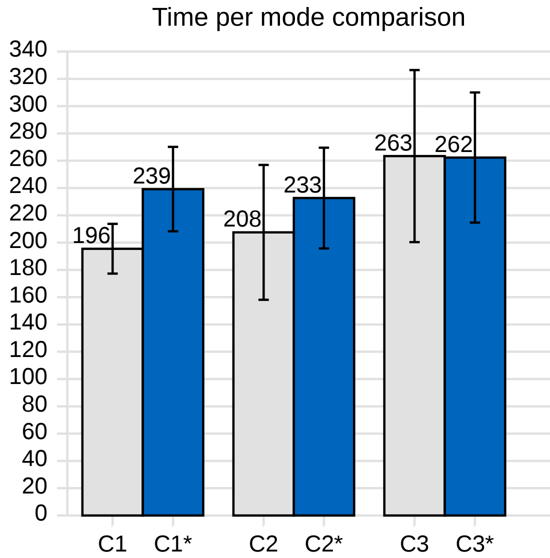


Figure 6.5.: Comparison of the durations per mode in study A and B.

feel certain in their decisions. Provided such a behavior, decision making would have taken longer in the second study.

6.4.3. RQ3. What is the relation between panorama and map usage?

This research question aims at analyzing the ratio between map and panorama usage. Therefore the number of interactions with the panorama image and the map preview, respectively, is counted. An interaction with the panorama image is a swipe on the screen in order to rotate the viewport. As the log data contained several items for long touch gestures (one item per second), subsequent entries at the same location are counted as a single interaction. Map interactions are touches on the screen area of the map in order to pan the map viewport. Map interactions are identified as unintended if the timestamps of the touch events are less than 500 milliseconds apart. Double-touches on the map area in order to expand or minimize the map are counted separately as toggle events.

All in all the interaction log data is rather sparse. In 17 of 36 runs no interaction can be identified. The rest shows at least one kind of interaction. Only those "interactive" runs will be considered here. Table 6.1 shows detailed data for all interaction techniques. Considering the average number of interactions users interacted mainly with the panorama surface. The interactions in order to pan the map occur second most frequently. The toggle interactions are used least frequently.

The interactions are equally distributed across the fully automatic and decision-point mode. 10 of the 19 interactive runs were recorded during the fully automatic mode. 53% of the panorama and

Table 6.1.: Summarized data for interaction comparison. An interaction occurrence is counted if at least one interaction happened during a run, i.e. multiple interactions of the same type and participant are aggregated. *Note*: values are calculated with data from "interactive" runs only.

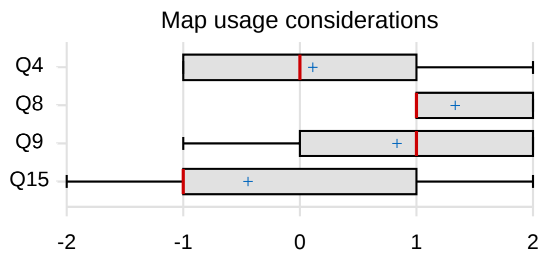
	Average number	SD	occurrences in "interactive" runs		
			total	$C1^*$	$C2^*$
panorama	3.53	5.69	15	8	7
map	2.68	6.26	10	6	4
toggle	1.68	1.85	8	6	2

60% of the map interactions happened in condition $C1^*$. An exception are the toggle interactions which seem to have occurred especially in the fully automatic mode (75%). To summarize, the qualitative data is little meaningful in terms of an interaction analysis and thus will not be considered anymore.

The second part of the map usage analysis is based on quantitative data. Following the subjects' responses to question four the map view alone is not entirely sufficient ($0.11, SD = 0.96$) but at least helpful for orientating ($1.33, SD = 0.49$). Moreover the automatic rotation of the map is considered convenient by most participants ($0.83, SD = 1.04$). Box plots in Figure 6.6a illustrate the map usage considerations. Asked if they used the panoramas or the map for orientating the self-estimation of users indicate that panoramas are used more frequently than the map view ($-0.44, SD = 1.20$). The answer categories here are "Mostly panoramas", "Slightly more panoramas", "Both equal", "Slightly more map" and "Mostly map". As the qualitative statement about the relation between panorama and map is rather weak, the responses of this question are analyzed in greater detail. 44% of users quote that they use the panoramas slightly more often whereas 22% state the opposite ("Slightly more map"). Two subjects use the panoramas as well as the map view. 17% answer that they use mostly panoramas for orientating while only 6% tell the same about their map usage. The histogram showed in Figure 6.7 illustrates the preferred use of panoramas.

An orientation technique based on 360° panoramas is not hampered by the map view. On a 5-step Likert scale from "never" to "frequently" users state that the map view rarely occluded important parts of panoramas ($-1.29, SD = 0.85$). Asked if they would have liked to be able to move the map area to another position on the screen subjects disagree ($-0.83, SD = 1.04$). Equally the participants do not want to resize the thumbnail map view, however the feature could be included in future versions as the result is not so clear ($-0.39, SD = 1.14$). A box plot visualization of the results is shown in Figure 6.6b.

Summary Analyzing the preferred method for orientating following the qualitative log data is hardly possible here. Many participants used the system without direct interaction. Thus, no



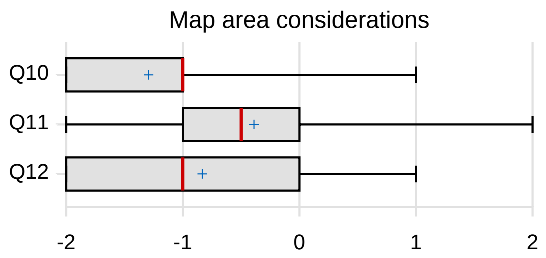
(a) Questionnaire entries considering map usage

Q4 The 2D map is sufficient for orientating

Q8 The 2D map was helpful for orientating.

Q9 The automatic rotation of the 2D map was convenient.

Q15 Please indicate the ratio whether you rather used the map or the panorama view



(b) Considerations regarding the map area UI

Q10 The map area occluded important parts of the panorama*

Q11 I would have liked to be able to resize the thumbnail map view

Q12 I would have liked to be able to move the thumbnail to another position on the screen

* One subject had to be excluded due to missing data

Figure 6.6.: Box plot visualization of map related responses.

favorite UI component can be identified objectively because the sparse log data delivers little meaningful statements. As a consequence the subjective self-estimation on the ratio between panorama and map usage of users has to be considered. A detailed look at the distribution of responses to the corresponding question (Q15) in the questionnaire of study B reveals that participants use mostly the panoramas for orientating. But there are as well users preferring the map preview. Hence, it would be advisable to integrate the possibility that users can choose their preferred kind of instruction presentation: panorama, maximized map or a split view of both.

6.4.4. RQ4. What are important locations for way instructions?

The manual mode in study B serves as a tool to identify those way points which are vital for indoor navigation. As highlighted above users have the possibility to request new instructions whenever they feel insecure about the correct way. Subjects are motivated to keep the number of requests as small as possible. This should ensure that they do not ask for updated instructions in situations where they can conclude the future way from obvious facts. E.g. intermediate requests on a long hallway without forks should become redundant. This setting aimed at verifying the hypothesis developed in Section 5.4.3 which illustrates that orientating only with panoramas of decision-points is as fast as finding one's way based on consequently updated panoramas. Moreover the gained results help to determine if the intermediate decision-points added in study B to all paths (green markers in Figure 6.1) are necessary or if users do not need them for orientating.

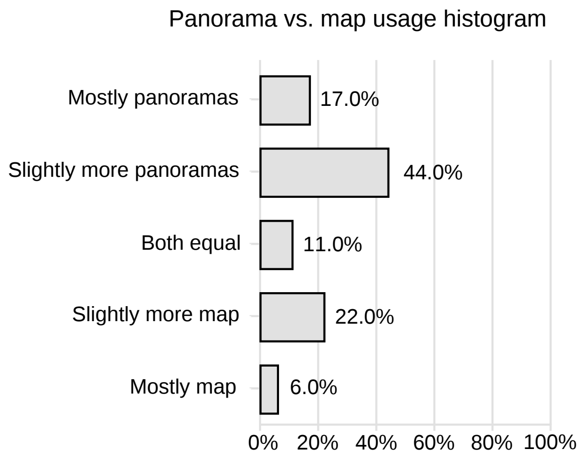
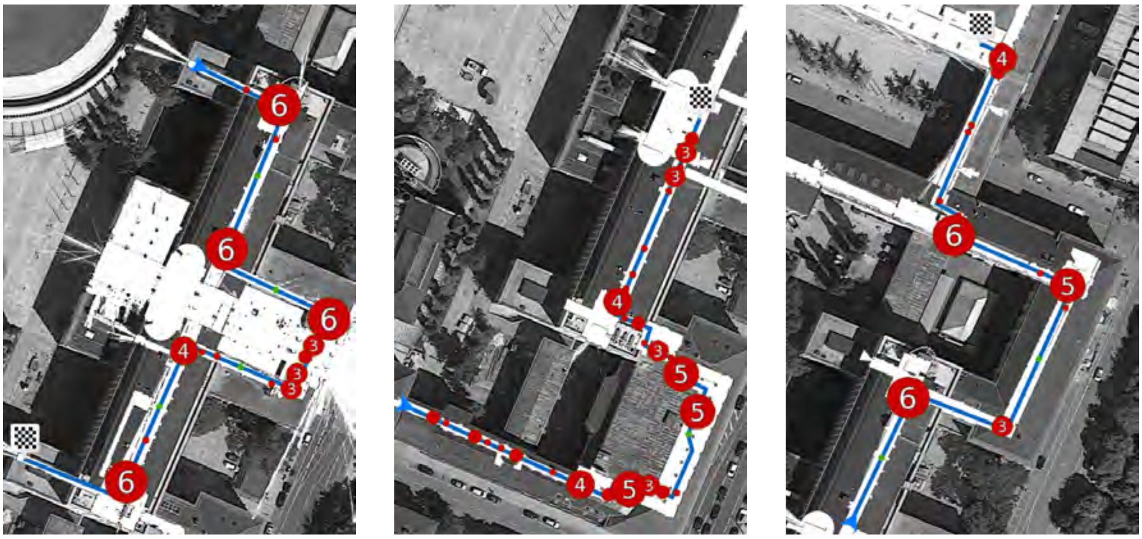


Figure 6.7.: Histogram showing the answer distribution of map and panorama usage.

Provided that decision-points are sufficient for indoor navigation, requests of users should only occur at crossings and forks of hallways, in halls with multiple exits and in confusing situations. Each time users require a panorama and instruction update the location of the request is logged. Hence, it is possible to determine the distance between the request position and the nearest decision-point. Before analyzing the data in detail the log is cleaned from subsequent requests of the same participant at the same location. These entries are considered to be duplicates resulting from repeated queries during the response time of the server.

In order to clarify which locations are important way points the number of requests per location is determined. This number ranges from 1 to maximum 6 due to the fact that the 18 participants used the manual mode on one of the three paths. The number of requests per location is illustrated in Figure 6.8. Each red circle is centered at a request location. The radius of each circle corresponds to the number of instruction requests sent from this position. Locations with a request number ≥ 3 are additionally annotated with a label containing the actual request count.

Figure 6.8a shows instruction requests on path A. It can be observed that at four locations all six participants queried a new instruction. These locations lie either short before or short after a path vertex and thus match the corresponding decision-point. At two other decision-points four requests and a double three combination of requests are detected. If the surrounding requests at those locations are considered as well, it gets clear that all subjects needed panorama and instruction updates at all decision-points of path A. In contrary, the additional intermediate decision-points added in the second study are not confirmed as important way points. An exception is the connection between the north and south part of the building where the path has a short outdoor section. Here five users were insecure about the further way and requested a new instruction before entering the south part of the building.



(a) Path A (b) Path B (c) Path C

Figure 6.8.: Manual instruction requests on each path in study B.

The analysis of the manual requests on path B is visualized in Figure 6.8b. The distribution of request locations is not as concentrated as on path A. But one can observe accumulations of requests mainly before or after path turns. Here the personal preferences of users, whether to request instructions right before or shortly after a turn, become obvious. Moreover the request analysis shows the unimportance of the subsequent zigzag decision-points. Users request updated instructions only at some of those decision-points and are able to find their way through the zigzag without further assistance. A location that should definitely be added as a decision-point to path B is just a short way apart from the goal. At this fork to another hallway six users want to know where they have to go. Hence, this additionally added intermediate decision-point is confirmed to be important. The other intermediate decision-points are not considered important. It is assumed that the many single requests at the beginning of path B are made due to the fact that the panorama images of this part of the building are very dark and users can not anticipate the further way.

The requests on path C (see Figure 6.8c) form a much clearer picture than those on path B. Taking into account single surrounding requests there are - similar to path A - accumulations of six requests matching four actual decision-point locations. The second decision-point from the start is only confirmed by three participants as there is no possibility to take another way. Equally the last two turns of the zigzag combination seem superfluous. As soon as users get the instruction to follow the narrow passageway to the bridge over the street no additional instructions are requested. Again the intermediate decision-points added specially for study B are not considered important as little to no queries were made at those locations.

In addition to the request per location analysis a matching between request and decision-point positions is conducted. If a request is sent directly from the location of a decision-point this is called a direct match. Other matches are called indirect matches and depend on a certain distance threshold. For each request location which is not a decision-point directly the distance Δ to the nearest decision-point is calculated. If Δ is smaller than the threshold t_i for level i there is a match between the request location r and the corresponding nearest decision-point dp. The Euclidean distance is used to determine Δ. Three threshold levels of $5, 5.5, 6, 7$ and 8 meters are defined. A match between a request location r and decision-point dp satisfies the following equation:

$$\sqrt{(r.x - dp.x)^2 + (r.y - dp.y)^2} < t_i, t_i \in \{5, 5.5, 6, 7, 8\}$$

The percentage of direct matches ranges from minimum 18% to maximum 83% per user. Averagely there are 49% direct matches for all participants. Applying the first threshold ($\Delta < 5$) leads to an average rate of 74% indirect matches. The per user minimum is 63% and there is one user who requested instructions only at decision-point locations which leads to a per user maximum of 100%. For all thresholds $t_i > 5$ the per user percentage of matches ranges between 64% and 100%. Thus, from now on, only the number of users matching 100% of decision-points with their requests are reported. At level two ($\Delta < 5.5$) 81% of the requests match a decision-point indirectly and the requests of two users match all decision-points. The next higher level ($\Delta < 6$) aggregates 85% of request-decision-point matches (six users with 100% matches). With a threshold of 7 meters half of the users have a 100% indirect matching rate. This corresponds with averagely 88% of requests matching decision-points. If the threshold is increased to 8 meters more than 90% of all requests can be classified as matches according to the above mentioned definition. At this level there are 10 users with all their requests being less than eight meters apart from decision-points. Figure 6.9 illustrates the matches per level while a detailed summary of the request to decision-point matching can be found in Appendix E.

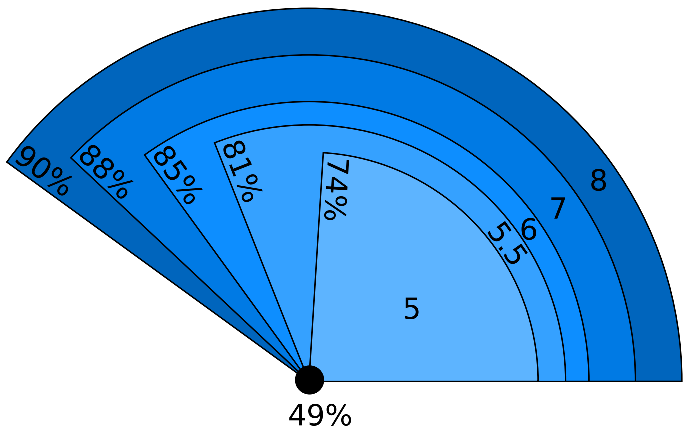
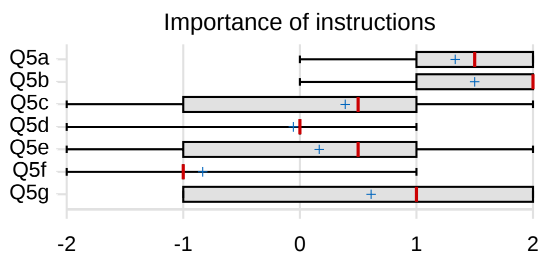


Figure 6.9.: Illustration of threshold size in meters and percentage of decision-point matches.

The importance of new instructions is evaluated quantitative as well. The questionnaire asks users to specify in which situations they find getting new instructions important. Seven different situations have to be rated on 5-step Likert scale from "Strongly disagree" to "Strongly agree". The box plots of the response distributions for the different question can be found in Figure 6.10. One can observe that users prefer getting new instructions right before $(1.33, SD = 0.77)$ or some meters before $(1.50, SD = 0.62)$ a possibility to choose a new path. But there are also subjects preferring new instructions after a turn $(0.39, SD = 1.24)$. This fits the findings from the above mentioned request location analysis, e.g. on path A (Figure 6.8a) where two request agglomerations (with three requests each) at the last but two decision-point can be observed right before and some meters before the turn. The analysis for path B indicates the different preferences whether receiving instructions before or after a turn is more important (Figure 6.8b). Subjects are indecisive about the question if they want to receive new instructions after they have not received instructions for a while $(-0.06, SD = 0.8)$. Asked if they would like getting new instructions when they slow down or accelerate users rather prefer updated instructions after slowing down $(0.17, SD = 1.10)$. In contrary they agree that new instructions are unnecessary when accelerating $(-0.83, SD = 0.71)$. Following the subjects' responses a situation better suited for new instructions is when they change their orientation, e.g. when they look around or turn backwards $(0.61, SD = 1.24)$.



a) Right before a possibility to choose another path
b) Some meters before a possibility to choose another path
c) Right after a turn
d) When I have not received an instruction for a while
e) When I slow down
f) When I accelerate
g) When I change my orientation (e.g. look back)

Figure 6.10.: In which situations would you find getting a new instruction important?

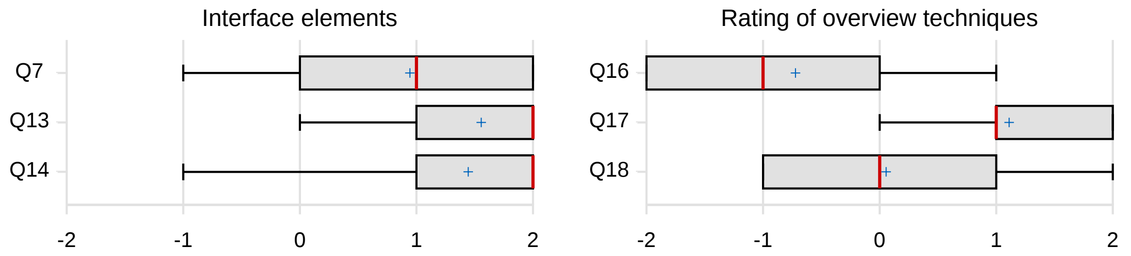
Summary The analysis of the manual requests shows that users find getting new way instructions at decision-points important. In a first step the analysis gives insights on agglomerations of instruction requests. These accumulations are mainly located near decision-points. It is found that some decision-points can be left out and that most intermediate decision-points are not considered important by users. A second step reveals that half of the request locations matched the position of a decision-point directly. 90% of all requests are located less than eight meters away from the nearest decision-point. Following a quantitative evaluation participants prefer getting new instructions before or after turns on the route. Users can also imagine to get instruction updates when slowing down or when changing the own orientation. Situations in which subjects accelerate or have not received instructions for a while are not considered important.

6.4.5. RQ5. What system elements do users consider as especially good or bad?

In the questionnaire users are asked to name at least one aspect of the system which they like most and least, respectively. Based on these answers and additional ordinal ratings it is worked out what system elements are advantageous and which should be thought over again. The aspects users liked most are for example the vibration feedback when new decision-point panoramas are shown, six users mentioned the thumbnail map, seven participants like the way to display route instructions with panoramas most and the distance information to the next decision-point and the goal is preferred by three subjects. Questions 13 and 14 in the questionnaire of study B confirm that the remaining distance indicator to the next decision-point $(1.56, SD = 0.70)$ respectively the goal $(1.44, SD = 0.86)$ are helpful tools. This and the fact that a button to request new instructions manually is considered important $(0.94, SD = 1.11)$ by users is displayed in Figure 6.11a.

The open-ended question about least liked system aspects result for example in negative remarks about the dark panoramas (three users complain about "night" views or the "too dark panoramas" being irritating). Two participants rate the distance indicator as too imprecise and two others think the same about the positioning of the arrow. Three subjects agree that the system requires too much attention and that it fixes one's mind on the device in an unfavorable way. That the number of panorama updates is too high in the fully automatic mode is confirmed by two participants. This mirrors as well in the responses to the question if the number of instructions was too high in $C1^*$ $(0.78, SD = 0.73)$. The number of instructions displayed in the decision-point mode seems to be convenient $(0.11, SD = 0.47)$. Any answer indicating that there are slightly too many instructions in condition $C2^*$ is assumed to refer to superfluous decision-points (like the second on path C or the many intermediate decision-points that were not confirmed in the analysis of Section 6.4.4).

Equally to users in study A users in study B think that seeing a list of all panoramas is not desirable $(-0.72, SD = 1.13)$. In contrary, an idea for future work could be a map displaying the entire route in advance to a way finding task. Users can imagine that such a tool could be helpful $(1.11, SD = 0.68)$. The vision of presenting users with two instructions at the same time is neither confirmed nor rejected $(0.06, SD = 1.16)$. How users rate these overview techniques can be observed in Figure 6.11b.



(a) Users rate the manual request button and the distance indicator positively.

Q7 The button to display a panorama of my current location with an instruction was important.

Q13 The remaining distance indicator until the next decision-point was helpful.

Q14 The remaining distance indicator until the goal was helpful.

(b) Ratings of different techniques providing an overview of way instructions.

Q16 Seeing a list of all panoramas in advance would have helped me.

Q17 Seeing a map with the entire path in advance would have helped me.

Q18 I would have liked to see more than one instruction at once.

Figure 6.11.: Considerations of good and bad system elements.

6.5. Discussion

The second study aims at evaluating the difference between the fully automatic mode and the decision-point mode. Both modes are enhanced by a map view displaying the current route. Additionally a bug discovered in the first study in the decision-point mode is eliminated. It is found that in terms of pleasantness the decision-point mode is rated better than the fully automatic mode. Moreover, the time needed to reach the destination is analyzed for both modes. This analysis shows no significant difference between the fully automatic mode and the decision-point mode. Thus, subsequently displaying new panoramas has no crucial advantage over displaying new instructions only at decision-points. The findings indicate that seeing more panoramas and thus having more information about the route is unnecessary as receiving new instructions at turns only is equally efficient and even better rated by users.

Results from the second questionnaire reveal that decision-points are sufficient for orientating. This is further proved by an additional test conducted during the second study. In a special condition, the manual mode, it is checked where users request new way instructions. As the participants do not know where their destination would be it is assumed that at least at every possibility to choose another path an instruction request would occur. The processing of the log data containing the exact locations of manual instruction requests leads to the insight that accumulations of instruction queries are located mainly around decision-points. Some existing decision-points can be removed from the path definitions as little requests have been made at those way points. A detailed matching of request locations to decision-points gives a coincidence of 49%. This means that half of the manual requests are made directly at decision-points. If requests lying within a radius of eight meters around decision-points are considered the matching rate increases to 90%. Hence, decision-points, i.e. way points where users have to decide whether to turn or not and locations with the possibility to take another path, are confirmed as important and crucial locations for indoor route descriptions.

In an effort to improve the way finding process with the existing application, a 2D map giving an overview of the route is provided. In the present study setting it is difficult to measure the preferred source of information. Whether the panoramas or the map view are preferred is tried to analyze based on the number of touches made to each component. But as nearly half of the users did not interact at least once with the device no meaningful statement is possible. A significant comparison of both kinds of presentation requires a further survey especially designed for finding user preferences between the panorama and the map view. Nevertheless, the results from the questionnaire based on a self-estimation of users about the ratio of panorama and map usage indicate that the panoramas are the main source of information. Presumably, the size of the bigger displayed panoramas is one reason for this preference. Additionally the panoramas contain more user related information as they mirror the real environment of the user. But there are also participants who liked the map most, probably because it puts the panoramas in a building-wide context. Thus the combination of indoor panoramas with a map preview of the route is a good compromise.

All in all the existing (e.g. the distance indicator) as well as the newly added UI components (like the map view) are rated as helpful. This disagrees with the findings of May et al. [27] who state that pedestrian navigation systems should not rely on the provision of distance information. However, user feedback during the survey runs, give rise to further improvements. For example, the coupling of the panorama and the map can be strengthened by rotating the user marker in the map according to the orientation of the panorama. Users turning the viewport of the panorama, e.g. by swiping horizontally over the screen, would then additionally recognize a rotation of the arrow representing their current location *and* orientation. Hornbæk et al. [40] prove the advantages of such coupling mechanisms. In the present prototype users missed the possibility to zoom the map in order to get a better overview of the route. Due to the inconsistent Internet connection in the building where the study took place and difficulties with offline map data it is not possible to provide a zoomable map at this point in time. As explained in Section 4.3.2 future prototypes, however, should incorporate the possibility to zoom the map.

Following user feedback the panorama presentation can be further improved, too. One participant noted that the panoramas could be rendered in a kind of gray-scale. Only significant objects, like fire extinguishers or eye-catching posters, should be left originally colorful. This forces users' focus to important parts of the picture and the highlighted objects simplify orientation. Following Elias and Sester [32] further annotating such emphasized landmarks is not necessary as users are able to infer the meaning of those colored regions on their own. Another interesting remark of a participant concerns the distance indicator and the direction arrow. The subject noted that the distance indicator, located at the bottom right, is out of one's focus. Hence, a possible improvement to the system is to display the distance information in the center of the screen. The direction arrow can be displayed with its outlines only and as users approach the next decision-point the area of the arrow is filled incrementally with color. The two ideas of the gray-scaled panoramas with highlighted objects and the distance-indicating arrow are illustrated in Figure 6.12.

(a) Highlighted wastebin. Rest of image is gray-scale transformed.

(b) Incrementally filled arrow indicates distance to next turn.

Figure 6.12.: Ideas for improvements after the second study.

The questionnaire of study B the questionnaire could be improved in three ways. First, the profession should not be polled in open-ended questions. Providing a list of predefined categories simplifies the analysis. Second, asking users about their sense of orientation and offering only two options is not fine grained enough. In this study 15 people state that they have a good sense of orientation while only three admitted that they don't. A 5-step scale would have fitted this question better. Third, the statement "The button to display a panorama of my current location with an instruction was important" is formulated badly. Users did not know which button was referred and the experimenter had to clarify that the question was about the importance of the possibility to request way instructions manually. Fortunately, these flaws affect only some general questions and it is not assumed that they have an influence on the answers of other items. Of course, these errors give rise to revise future questionnaires once more.

Chapter 7.

Interpretation

This section provides an overview of the initial (Section 5) and the follow-up study (Section 6). Findings of both studies are compared and common results are highlighted. The differences and similarities are interpreted and put into the context of improved instruction presentations. The section outlines experiences made during both studies and gives ideas for enhancements in future prototypes.

In the initial study it is showed clearly that the fully automatic mode is the users' favorite mode of instruction presentation. The continuously updated instructions make users feel save at every location of the path. The locations displayed in the decision-point mode are found to lack context, especially if far away or obscured by obstacles. Thus, guidance abilities of the fully automatic mode are rated better. The manual mode requires users to update the way instruction on their own. This makes interaction and orientation necessary, just walking along a displayed route is not possible. It is found that the manual mode bears advantages especially for people with a good sense of direction and those who already know a building vaguely.

The follow-up study provides participants with an extra kind of instruction presentation. A 2D map showing the route, the next instruction and the user's position as well is integrated into the fully automatic and decision-point modes. The latter becomes the favorite mode of most users in the second study. The ratings in terms of pleasantness and guidance raise for the decision-point mode and fall for the fully automatic mode. The differences of the ratings of the decision-point mode differ significantly between study A and B. Considering the fully automatic mode the users are significantly less pleased in study B. The ratio of people who would choose the decision-point or the fully automatic mode is completely opposed between the two studies. While 82% favor the fully automatic mode in the first study, in the second 76% would choose the decision-point mode.

Concluded, the presented findings from the initial and the follow-up study tell that presenting instructions at decision-points of the route supported by spatial information from a map rather satisfies users than providing them with the same information and even more instructions. The decision-point mode displaying instructions at lower frequency requires less cognitive resources at the cost of less direction information. All the more remarkably is that using the decision-point mode is not significantly slower.

Furthermore, the gained insights highlight the possibility of the map presentation to help users in confusing and unclear situations where the future way is difficult to anticipate. Such situations are winding, zigzag like path segments, parts of the route where only underexposed pictures are available and decision-points localized far apart from each other. Having a map at hand showing the actual route and the own location simplifies navigation in those situations.

Study B shows that interpreting the additional information takes extra time. Participants using the both instruction modes including the map view are found to be averagely slower than participants using the same modes in the first study. For the fully automatic mode this difference is proved to be statistically significant. This gives rise to the assumption that the cognitive capacities are overstrained by the combination of continuously updated panoramas and map markers. The compulsion to frequently match the newly displayed panoramas to the real environment and the spatial analysis of the relationship between the own position and the displayed map information seems to be too much load for users. Therefore, they need longer to process the presented instructions and decision making is delayed. Future prototypes can incorporate the possibility to hide the map presentation in order to satisfy the preferences of different users. So users preferring the fully automatic mode can hide the map and use their preferred mode as fast as study A implied.

The difference in time is not found to be significant for the decision-point mode. However, the results from the second study indicate a higher average duration with the enhanced instruction modes. But, in case of the decision-point mode, the system is rated higher in terms of how pleasing the participants perceived it. This leads to the question what matters most: satisfaction or speed? The answer to this question is up to a repeated evaluation with correspondingly adjusted settings.

User feedback from the first study indicated the need for intermediate decision-points. These close the gap between decision-points being far apart from each other. The instruction at those intermediate points is to follow the hallway straight ahead. Thus they are not crucial for orientating but help navigating as they give additional confidence. The second study aimed at evaluating the need for such helper-points. Regarding the decision-point condition the improved ratings indicate their usefulness. But according to some individual users decision-points displaying a "go ahead" instructions (in order to confirm the right way) are distracting and thus are named in the questionnaire as negative aspects of the system. Equally the data from the manual condition in study B suggest that there is no desperate need for such intermediate helper-points.

In contrary to the intermediate decision-points the feature to request the immediately next instruction at one's current position is found to be a well accepted helper. Either the re-localization method or the manual request button in the both versions of the manual mode is proved to be important and convenient to use. The manual mode is rated least pleasing in study A, however, user feedback written in the questionnaire or vocalized during the evaluation tasks indicates that the satisfaction with the automatic modes can be further improved when providing a possibility to request instructions manually. For example one participant asked if there is a "repeat last step" feature in the fully automatic mode. A manual intervention possibility into the automatic instruction sequence seems to be an intuitive opportunity to serve different user preferences.

Another point confirmed in both user studies is the significance of high quality reference images in a VR-based indoor navigation system. "High quality" has different meanings for the image matching process of computer algorithms or humans. While feature extraction algorithms require the reference images to be free of inferring factors like people walking through the scene, participants in both studies emphasized the importance of daytime pictures in the VR-to-reality matching process. The present reference image set was captured by night to ensure the recording of as many as possible original building features. But, for example, this implies underexposed images of hallways behind closed glass doors where the lightning is switch off. Moreover, reference-points in the real environment outside the building observable through windows are difficult to identify in the nightly VR scene. As proper exposed pictures are advantageous for the algorithmic feature extraction, too, it is suggested to capture reference datasets at daytime when the building is closed, e.g official holidays or feast days, and no people disturb the recording process.

Further common feedback from both studies additionally concerns the reference images. In the first as well as in the second study individual participants suggest to highlight salient objects in the panorama view. The comments include e.g. highlighting fire-extinguishers with a glowing border or leaving dustbins or posters colored while the rest of the image is displayed in greyscales. As outlined in Section 2.2.2 such visually emphasized objects are important landmarks and support pedestrian navigation outdoors as well as indoors.

Well-known from car navigation systems are spoken way instructions. This concept of presenting users with instructions can be adopted for indoor navigation tasks. In both studies users indicate that they would like to hear instructions like "In the hallway take the third door on the left. Then follow the corridor to the entrance hall". Considering privacy one participant suggested to use headphones to listen to the instructions. Thus other pedestrians are not molested. He further suggests to put the mobile device into a pocket of one's jacket in order to use the navigation system incognito without publishing the own need for help. Leaving aside the fact that a vision-based system won't work inside pockets the feedback shows the different aspects of privacy considerations.

Experiences during the *wizard-of-oz* evaluations show how difficult it can be to guarantee consistent test conditions in a laboratory settings controlled by the experimenter. Location updates have to be sent manually and at the same position for all participants, conversations during or between the different tasks may not influence the user's perception and assisting subjects with answering the questionnaire or with understanding the briefing has to be as objective as possible. Else participants could adapt to the experimenter's opinion or they could adjust their rating in order to please.

Chapter 8.

Conclusion and Outlook

In this thesis approaches for improving instruction presentations for indoor navigation are presented. The underlying concepts are explained and the enhancements applied to an existing application are motivated. Three different implementations of instruction modes are iteratively improved. Two user studies are conducted to evaluate a fully automatic, a decision-point and a manual mode. User feedback from the initial study is analyzed and the instruction modes are adjusted accordingly. Concepts evaluated as being advantageous for pedestrian navigation in related works are considered as well and integrated in the modes. Thus, in the second study a map view is evaluated besides the panoramic instruction presentation.

The first study shows that the fully automatic mode is favored by users. It is rated best in terms of overall pleasantness and guidance to the goal. In average, the fully automatic mode is found to be faster than the other modes. Using the manual mode it takes averagely longest to reach the destination. Regarding time the decision-point mode differs not significantly from the fully automatic mode and users rate panoramas at decision-point as being sufficient for orientating. Especially in combination with a concept called re-localization which provides users with instructions of their actual position indoor navigation based on decision-points only is found to be a promising approach. The initial study further highlighted that extensive manual interaction in the wayfinding tasks is not favored by users. Though, the manual mode revealed usage patterns that can be assigned to two user groups characterized by their abilities to orientate.

In a follow-up study the enhancements, e.g. the map view, made to the modes after the initial study are evaluated. Furthermore, the statements of the first study are verified. The second study is designed to evaluate the fully automatic and the decision-point mode. The manual mode is used to track where users would find getting instructions important. Regarding the map view it becomes clear that providing way instructions on a map is not sufficient for indoor navigation. Nevertheless, the map view is found to be helpful. The ratio between panorama and map usage depends on personal preferences. Also the locations rated important for getting new instruction vary from user to user. But accumulations of requests for instruction are located near decision-point locations.

Included in the findings of the second study are the inverted user ratings for the fully automatic and the decision-point mode. Participants in the follow-up study favor the latter. Obviously the spatial map instructions add contextual information of the building structure to the panorama pictures and thus overcome shortcomings of the decision-point mode. Considering time it takes equally long to reach the goal with both modes. Continuous instruction updates like in the fully automatic mode are not found to be more efficient nor more satisfying. Similarly to the first study, panoramas of turn locations displayed in the decision-point mode are confirmed to be sufficient for orientating. Following the insights of both studies the main finding of this thesis is that indoor navigation based on instruction presentation at decision-points only is equally efficient as indoor navigation based on continuous instruction presentations. This is true in terms of time need to reach the goal and especially in terms of user satisfaction.

Interface elements adopted from existing work are repeatedly evaluated as helpful and convenient to use. This thesis contributes in further improving these presentation and interaction tools. The feedback of both conducted studies is summarized and the interpretations leading to new ideas and suggestions for enhanced features are outlined. For example the concept of an instruction presentation in form of an arrow including information about walking direction *and* distance is highlighted.

As outlook to future versions the presented indoor navigation system should be based on a model of a building-wide path network. Extending the existing path structure to a mature graph with hallways as edges and decision-point as vertices would offer the possibility to calculate the shortest or fastest way between two locations in the building. The resulting implications on users are an interesting field of research. Do pedestrians always favor the fastest way? Is the shortest way always preferred even if it includes stairs? Generally, evaluations including paths on multiple floors seem necessary. For example, the presentation of overlapping hallways on different floors in the 2D map could be analyzed and a convenient way of visualizing the instruction arrow pointing up or down a staircase has to be found.

A thorough path-location network could also be the base for an automatic decision-point retrieval. Similar to rating algorithms for Internet pages decision-point vertices in the network can be assigned with an "importance value". Thus, important decision-points are included in the route presentation in any case whereas unimportant decision-points are excluded. Such a rating could be supported by manual requests from users. Locations with many user requests would receive an increased "importance value" and the rating of locations with little request would be decreased over the time.

The feedback on the fully automatic mode indicates that users feel somehow forced to look at the continuously updated instructions and just walk along the indicated route without noticing their surrounding. Self-reflecting their behavior participants worried about the fact that human abilities like the sense of direction are consequently replaced by machines. An outlook can be to include persuasive elements in (indoor) navigation systems animating users to learn and to keep basic navigational abilities. In a LBS context, for example, at distinct locations users could receive a task to locate their actual position on a street or building map. This would require re-orientation and users would perceive their environment consciously. From an economic point of view successfully mastered tasks could be rewarded with vouchers making shops attractive which users might have missed with all their attention bound by the mobile device.

At the end of the questionnaire all participants are asked if they could imagine to use the presented or a similar system in real situations. Besides multiple scenarios where to apply indoor navigation systems like hospitals, airports, shopping malls and office or administration buildings the majority of users quoted to use indoor navigation systems. Provided that the necessary data is free of charge, easily and freely accessible indoor navigation systems enjoy great popularity.

Appendix A.

Initial Study - Briefing

Indoor Navigation Survey - Briefing

First of all we want to thank you for taking part in this survey! You will evaluate a vision-based navigation system that uses the camera of the smartphone to determine your location.

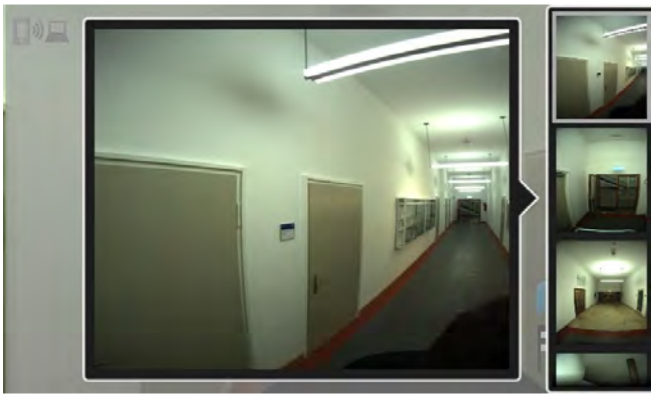
You will test the different functionalities of the application in several situations. Remember that it isn't yourself who is tested, but the system. So use the system as you want and don't be afraid of making mistakes.

The application will guide you along different paths. Therefore it displays 360° panorama views at different locations in the building (similar to Google Street View). An arrow will help you finding the right walking direction. By swiping left and right on a panorama you can turn the view and look around at the selected location. At the bottom right you find information about how far away the goal or the next turn is.

You will use three different modes:

1. Automatic, panoramas of your current location
2. Automatic, panoramas of locations where you have to turn
3. Manual, panoramas of locations where you have to turn

In mode 1 you will receive the panoramas of your current location automatically. All you have to do is to walk along the displayed route. In mode 2, you will receive automatically the panorama of the location where you have to make your next turn (decision point). As soon as you have passed this point, you receive the panorama of the following decision point.



In mode 3, you likewise have the panoramas of decision points available, but they will not change automatically. You can swipe up or down to see the previous or next panorama in the list of instructions. You can also pull or click the small handle on the right to enable the panorama preview. You can scroll through the previews and select one by clicking on the large preview image. If you feel lost and want the system to relocalize yourself, raise up the phone so that the camera can see the environment and determine your location again. The system will display the panorama showing your correct next instruction.

Please don't hesitate giving your impressions on the system during the runs.

At the end of the survey you will be asked to fill out a short questionnaire.

Do you have any questions?

Appendix B.

Initial Study - Questionnaire

Questionnaire

General Questions

1. Gender

   O male O female

2. Date of Birth

   ____

3. Do you own a smart phone?

   O yes O no

4. If yes, how often do you use navigational applications?

Never	Rarely	Sometimes	Often	Frequently
O	O	O	O	O

5. What kinds of navigational systems do you know? (multiple choice)

   - O 2D (simple map)
   - O 2.5D (some car navigation systems)
   - O 3D (looks like a video game)
   - O Augmented Reality
   - O other

   ____
   ____
   ____
   ____

6. Do you have experience with indoor navigation?

   O yes O no

Main Questions

1. I found the method pleasing to use

Automatic mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Automatic mode (decision points only)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Manual mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

2. I felt guided well to the goal

Automatic mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Automatic mode (decision points only)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Manual mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

3. Which mode of navigation would you choose to use? (single choice)

O Automatic mode
O Automatic mode (decision points only)
O Manual mode

4. Panoramas of decision points are sufficient for orientating.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

5. Changing panoramas by swiping up and down is useful. (manual mode only)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

6. Changing panoramas in the list view is useful. (manual mode)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

7. The ability to re-localize myself is useful.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

8. The method of moving up the phone to re-localize is convenient.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Final Questions

1. What kinds of usages for indoor navigation could you imagine?

____
____
____

2. Are there other applications you would like the system to share information with (e.g. calendar application)

____
____
____

3. There should be the possibility to interact with real-world objects (e.g. timetables)?

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
o	o	o	o	o

4. I would appreciate if the system could suggest navigation goals (e.g. in an airport or shopping mall)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
o	o	o	o	o

5. Which aspects of the system did you not like / did you have difficulties with (if any)

____
____
____

6. Which aspects of the system did you like (if any)

____
____
____

7. Do you have further feedback or comments?

____
____
____

Appendix C.

Follow-Up Study - Briefing

Indoor Navigation Survey - Briefing

First of all we want to thank you for taking part in this survey! You will evaluate a vision-based navigation system that uses the camera of the smartphone to determine your location.

You will test the different functionalities of the application in several situations. Remember that it isn't yourself who is tested, but the system. So use the system as you want and don't be afraid of making mistakes.

The application will guide you along different paths to a destination you will not know in advance. You only have to follow the instructions of the system. To guide you the system displays 360° panorama views at different locations in the building (similar to Google Street View). An arrow will help you finding the right walking direction (1). By swiping left and right on a panorama you can turn the view and look around at the selected location (2). At the bottom right you find information about how far away the goal and the next turn is (3). In the upper right corner you see a 2D map of the path (4).

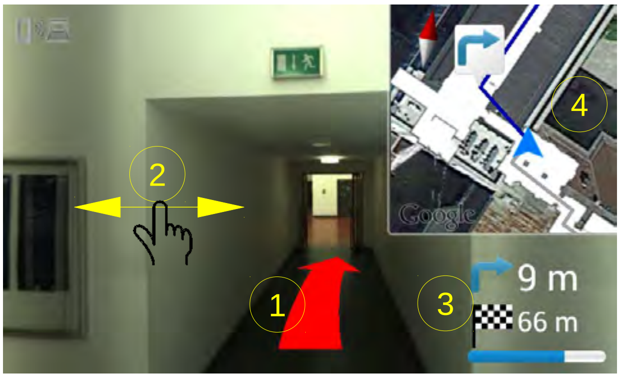


Illustration 1: Preview of mode A

You will use three different modes:

- A <u>Automatic:</u> Continuous panorama updates of your current location
- B <u>Automatic (decision points):</u> Only panoramas of locations where you have to turn
- C <u>Manual:</u> You will help the system to find important turn locations

In mode *A* you will receive panoramas with embedded navigation instructions automatically every few meters. All you have to do is to walk along the displayed route. In mode *B*, you will receive panoramas in larger intervals and of significant points (e.g. when you have to make a turn). As soon as you have passed this point, you receive the panorama of the following decision point.

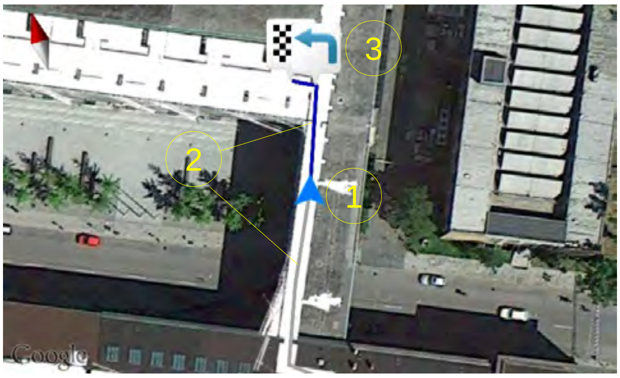


Illustration 2: Map preview expanded

The 2D map displays the current path and updates your location (1) continuously. The gray line indicates the part of the path you already passed while the blue one represents your further way (2). The map's orientation changes as the orientation of the phone changes. Furthermore markers of the next turn location and the goal are displayed (3). You may move the map by simply dragging it and expand/minimize it by double tapping it.
In mode *C*, you will not receive any panorama update except you ask for it. When you feel that you need a new instruction and want the system to display your current location and the next instruction simply press the button. You see a counter of your location requests below the button. Only request an instruction when you really need it, i.e. try to make as little requests as possible!

Please don't hesitate to give your impressions on the system during the runs.

At the end of the survey you will be asked to fill out a questionnaire. Do you have any questions?

Appendix D.

Follow-Up Study - Questionnaire

Questionnaire

General Questions

1. Gender — O male O female
2. Date of Birth ___.___._____
3. Do you own a smart phone? — O yes O no
4. If yes, how often do you use navigational applications?

Never	Rarely	Sometimes	Often	Frequently
O	O	O	O	O

5. What kinds of navigational systems do you know? (multiple choice)

   - O 2D (simple map)
   - O 2.5D (some car navigation systems)
   - O 3D (looks like a video game)
   - O Augmented Reality
   - O other

   ____

6. Do you have experience with indoor navigation? — O yes O no
7. Your profession? ____
8. Are you left-handed? — O yes O no
9. Are you familiar with the building we are in? — O yes O no
10. Do you consider yourself to have a good sense of orientation? — O yes O no

Main Questions

1. I found the method pleasing to use

Automatic mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Automatic mode (decision points only)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

2. I felt guided well to the goal

Automatic mode

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Automatic mode (decision points only)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

3. Which mode of navigation would you choose to use? (single choice)

   - O Automatic mode
   - O Automatic mode (decision points only)

4. The 2D map is sufficient for orientating

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

5. Panoramas of decision points are sufficient for orientating.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

6. The number of instructions was

Automatic mode

Way too little	Slightly too little	Convenient	Slightly too high	Way too high
O	O	O	O	O

Automatic mode (decision points only)

Way too little	Slightly too little	Convenient	Slightly too high	Way too high
O	O	O	O	O

7. The button to display a panorama of my current location with an instruction was important.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

8. The 2D map was helpful for orientating.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

9. The automatic rotation of the 2D map was convenient.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

10. The map area occluded important parts of the panorama

Never	Rarely	Sometimes	Often	Frequently
O	O	O	O	O

11. I would have liked to be able to resize the thumbnail map view

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

12. I would have liked to be able to move the thumbnail to another position on the screen

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

13. The remaining distance indicator until the next decision point was helpful

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

14. The remaining distance indicator until the goal was helpful.

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

15. Please indicate the ratio whether you rather used the map or the panorama view

Mostly panoramas	Slightly more panoramas	Both equal	Slightly more map	Mostly map
O	O	O	O	O

16. Seeing a list of all panoramas in advance would have helped me

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

17. Seeing a map with the entire path in advance would have helped me

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

18. I would have liked to see more than one instruction at once

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Final Questions

1. What kinds of usages for indoor navigation could you imagine?

___

___

___

2. Imagine the navigation system would be working in the above named situations. Would you use it?

___

___

___

3. Name at least one aspect of the system you liked most

___

___

___

4. Name at least one aspect of the system you liked least

___

___

___

5. In which situations would you find getting a new instruction important?

Right before a possibility to choose another path

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Some meters before a possibility to choose another path

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Right after a turn

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

When I have not received an instruction for a while

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

When I slow down

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

When I accelerate

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

When I change my orientation (e.g. look back)

Strongly disagree	Disagree	Neutral	Agree	Strongly agree
O	O	O	O	O

Other(s):

6. Do you have further feedback or comments?

Appendix E.

Follow-Up Study - Decision-Point Matching Analysis

The following table shows detailed results from the matching of request locations to actual decision-points. For each participant the path on which the manual mode was used in study B is named and the identifiers of the request locations are listed. The identifiers are shortened, the raised number denotes the dataset of the location (^{1}: ground floor 1/2, ^{2}: ground floor 2/2, ^{11}: 1st floor, ^{12}: 1st floor N1). Right to each identifier the distance to the nearest decision-point is found. Entries with $\Delta = 0$ specify direct matches as defined in Section 6.4.4. The lower part of the table shows the total count of requests and the percentage of matches at a distinct threshold level. In total 149 requests are made. Numbers in parenthesis define the absolute number of matches. The last column of the lower part shows the percentage of matches for all participants.

Participant	1		2		3		4		5		6		7		8		9	
Path	3	Δ	2	Δ	1	Δ	1	Δ	3	Δ	2	Δ	2	Δ	1	Δ	3	Δ
Requests	1181^{11}	0	512^{11}	13.1	085^{1}	0	085^{1}	0	1181^{11}	0	481^{11}	12.6	476^{11}	8.5	030^{1}	9.7	1136^{11}	0
	1284^{11}	0	571^{11}	0	143^{1}	0	143^{1}	0	1284^{11}	0	556^{11}	4.6	517^{11}	9.0	085^{1}	0	1181^{11}	0
	1375^{11}	6.6	622^{11}	0	209^{1}	5.6	209^{1}	5.6	1375^{11}	6.6	609^{11}	0	528^{11}	0	107^{1}	9.9	1342^{11}	0
	1417^{11}	4.1	634^{11}	4.3	308^{1}	4.0	308^{1}	4.0	1417^{11}	4.1	622^{11}	0	544^{11}	11.6	143^{1}	0	1417^{11}	4.1
	1426^{11}	0	704^{11}	0	572^{2}	4.7	572^{2}	4.7	876^{12}	0	627^{11}	3.1	556^{11}	4.6	209^{1}	5.6	858^{12}	4.3
	902^{12}	22.0	745^{11}	0	609^{2}	4.4	591^{2}	5.1	874^{12}	2.2	662^{11}	4.0	634^{11}	4.3	308^{1}	4.0	851^{12}	2.7
	854^{12}	0	870^{11}	5.0	335^{2}	13.2	481^{2}	5.2	854^{12}	0	704^{11}	0	699^{11}	4.2	572^{2}	4.7		
			922^{11}	0	320^{2}	0	320^{2}	0			744^{11}	0	761^{11}	4.3	585^{2}	1.4		
			996^{11}	7.6							745^{11}	0	820^{11}	0	617^{2}	0		
											767^{11}	0	893^{11}	20.8	320^{2}	0		
											775^{11}	0	1002^{11}	4.3				
											815^{11}	4.2						
											870^{11}	5.0						
											876^{11}	8.4						
											986^{11}	3.1						
											1002^{11}	4.3						
Sum	7(7)		9(9)		8(8)		8(8)		7(7)		16(19)		11(11)		10(10)		6(6)	
$\Delta = 0$	57% (4)		56% (5)		38% (3)		38% (3)		57% (4)		44% (7)		18% (2)		40% (4)		50% (3)	
$\Delta < 5$	71% (5)		67% (6)		75% (6)		63% (5)		86% (6)		81% (13)		64% (7)		70% (7)		100% (6)	
$\Delta < 5.5$	71% (5)		78% (7)		75% (6)		88% (7)		86% (6)		88% (14)		64% (7)		70% (7)		100% (6)	
$\Delta < 6$	71% (5)		78% (7)		88% (7)		100% (8)		86% (6)		88% (14)		64% (7)		80% (8)		100% (6)	
$\Delta < 7$	86% (6)		78% (7)		88% (7)		100% (8)		100% (7)		88% (14)		64% (7)		80% (8)		100% (6)	
$\Delta < 8$	86% (6)		89% (8)		88% (7)		100% (8)		100% (7)		88% (14)		64% (7)		80% (8)		100% (6)	

Participant	10		11		12		13		14		15		16		17		18		
Path	3	Δ	2	Δ	1	Δ	1	Δ	3	Δ	2	Δ	2	Δ	1	Δ	3	Δ	
Requests	1181^{11}	0	622^{11}	0	085^{1}	0	085^{1}	0	1181^{11}	0	476^{11}	8.5	522^{11}	4.7	003^{1}	0	1181^{11}	0	
	1284^{11}	0	640^{11}	0	143^{1}	0	143^{1}	0	1375^{11}	6.6	512^{11}	13.1	556^{11}	4.6	085^{1}	0	1291^{11}	2.0	
	1375^{11}	6.6	704^{11}	0	209^{1}	5.6	209^{1}	5.6	1417^{11}	4.1	528^{11}	0	622^{11}	0	143^{1}	0	1375^{11}	6.6	
	1417^{11}	4.1	745^{11}	0	304^{1}	0	583^{2}	0	858^{12}	4.3	556^{11}	4.6	634^{11}	4.3	209^{1}	5.6	1387^{11}	12.6	
	854^{12}	0	870^{11}	5.0	583^{2}	0	481^{2}	5.2	851^{12}	2.7	571^{11}	0	704^{11}	0	304^{1}	0	1417^{11}	4.1	
			922^{11}	0	481^{2}	5.2	320^{2}	0			622^{11}	0	745^{11}	0	583^{2}	0	854^{12}	0	
					320^{2}	0					640^{11}	0	767^{11}	0	481^{2}	5.2			
											704^{11}	0	870^{11}	5.0	320^{2}	0			
											745^{11}	0	916^{11}	4.5					
											767^{11}	0	996^{11}	7.6					
											815^{11}	4.2							
											881^{11}	12.7							
											922^{11}	0							
											996^{11}	7.6							
Sum	5(5)		6(6)		7(9)		6(6)		5(5)		14(14)		10(10)		8(8)		6(6)		149
$\Delta = 0$	60% (3)		83% (5)		71% (5)		67% (4)		20% (1)		57% (8)		40% (4)		75% (6)		33% (2)		49%
$\Delta < 5$	80% (4)		83% (5)		71% (5)		67% (4)		80% (4)		71% (10)		80% (8)		75% (6)		67% (4)		74%
$\Delta < 5.5$	80% (4)		100% (6)		86% (6)		83% (5)		80% (4)		71% (10)		90% (9)		88% (7)		67% (4)		81%
$\Delta < 6$	80% (4)		100% (6)		100% (7)		100% (6)		80% (4)		71% (10)		90% (9)		100% (8)		67% (4)		85%
$\Delta < 7$	100% (5)		100% (6)		100% (7)		100% (6)		100% (5)		71% (10)		90% (9)		100% (8)		83% (5)		88%
$\Delta < 8$	100% (5)		100% (6)		100% (7)		100% (6)		100% (5)		79% (11)		100% (10)		100% (8)		83% (5)		90%

Appendix F.

Log File Structure

Table F.1.: Detailed log data structure and possible values for each field.

Column	Information	Values
A	Absolute time	milliseconds since 1/1/1970
B	Relative time	milliseconds since start of instruction mode
C	Event type	Event that triggered the log entry 0: new location 1: new location by user 2: server disconnected 3: initialization 4: user request 5: map toggle 6: map touch 7: panorama touch
D	Message type	Type of server-client/client-server message 1: heartbeat 2: location 3: text 4: initialization 5: finish 6: user request 7: wrong way warning
E	Location ID	Dataset identifier for the current position
F	Preview Location ID	Dataset identifier for the currently displayed location
G	Path ID	Path identifier 1: path A 2: path B 3: path C 9: demo path
H	Measurement	Unit of the distance information Meters Imperial
I	Orientation mode	How the panorama is rotated 0: Touch 1: Touch + go back 2: Sensor
J	Instruction mode ID	Mode identifier 0: fully automatic 1: decision-point 2: manual
K	Experimenter	Whether or not the message was sent by the experimenter True False

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface

AR Augmented Reality

BMW Bayerische Motoren Werke

BPN BMW Personal Navigator

CSV Comma Separated Values

DECT Digital Enhanced Cordless Telecommunication

GIMP GNU Image Manipulation Program

GPS Global Positioning System

GUI Graphical User Interface

IDE Integrated Development Environment

IMU Inertial Measurement Unit

IR infrared

LBS Location Based Services

MARS Mobile Augmented Reality System

OS Operating System

PBVE Panorama-Based Virtual Environment

PCA Principal Component Analysis

PDA Personal Digital Assistant

SIFT Scale Invariant Feature Transform

SURF Speeded-Up Robust Features

TUM Technische Universität München

UI User Interface

URL Uniform Resource Locator

UTM Universal Transverse Mercator

VMI Fachgebiet Verteilte Multimodale Informationsverarbeitung

VR Virtual Reality

WGS World Geodetic System

WIM World In Miniature

WLAN Wireless Local Area Network

ZUI Zoomable User Interface

Bibliography

[1] H. Hile, R. Vedantham, G. Cuellar, A. Liu, N. Gelfand, R. Grzeszczuk, and G. Borriello, "Landmark-based pedestrian navigation from collections of geotagged photos," in *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, MUM '08, (Umeå, Sweden), pp. 145–152, ACM, December 2008.

[2] A. Mulloni, H. Seichter, and D. Schmalstieg, "Handheld augmented reality indoor navigation with activity-based instructions," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, (Stockholm, Sweden), pp. 211–220, ACM, August 2011.

[3] J. Baus, A. Krüger, and W. Wahlster, "A resource-adaptive mobile navigation system," in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, (San Francisco, CA, USA), pp. 15–22, ACM, January 2002.

[4] M. Werner, M. Kessel, and C. Marouane, "Indoor positioning using smartphone camera," in *International Conference on Indoor Positioning and Indoor Navigation*, IPIN '11, (Guimarães, Portugal), pp. 1–6, IEEE, September 2011.

[5] V. Zeimpekis, G. M. Giaglis, and G. Lekakos, "A taxonomy of indoor and outdoor positioning techniques for mobile location services," *SIGecom Exchanges*, vol. 3, no. 4, pp. 19–27, 2002.

[6] M. Werner, *Ubiquitous Navigation*. PhD thesis, Ludwig-Maximilians-Universität München, 2012.

[7] A. Butz, J. Baus, A. Krüger, and M. Lohse, "A hybrid indoor navigation system," in *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, (Santa Fe, NM, USA), pp. 25–32, ACM, January 2001.

[8] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2 of *INFOCOM '00*, (Tel Aviv, Israel), pp. 775–784, IEEE, March 2000.

[9] J. Hightower, R. Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," tech. rep., UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, USA, February 2000.

[10] M. Kranz, C. Fischer, and A. Schmidt, "A comparative study of DECT and WLAN signals for indoor localization," in *IEEE International Conference on Pervasive Computing and Communications*, PERCOM '10, (Mannheim, Germany), pp. 235–243, IEEE, April 2010.

[11] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, (Seoul, Korea), pp. 114–123, ACM, September 2008.

[12] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[13] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.

[14] H. Huang and G. Gartner, "A survey of mobile indoor navigation systems," in *Cartography in Central and Eastern Europe* (G. Gartner and F. Ortag, eds.), Lecture Notes in Geoinformation and Cartography, pp. 305–319, Springer Berlin Heidelberg, 2010.

[15] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[16] D. G. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2 of *ICCV '99*, (Kerkyra, Greece), pp. 1150–1157, IEEE, September 1999.

[17] H. Bay, T. Tuytelaars, and L. Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer Berlin Heidelberg, 2006.

[18] H. Hile and G. Borriello, "Positioning and orientation in indoor environments using camera phones," *IEEE Computer Graphics and Applications*, vol. 28, no. 4, pp. 32–39, 2008.

[19] P. Liu, X. Sun, N. D. Georganas, and E. Dubois, "Augmented reality: A novel approach for navigating in panorama-based virtual environments (PBVE)," in *The 2nd IEEE Internatioal Workshop on Haptic, Audio and Visual Environments and Their Applications*, HAVE '03, (Ottawa, Ontario, Canada), pp. 13–18, IEEE, September 2003.

[20] H. Kawaji, K. Hatada, T. Yamasaki, and K. Aizawa, "Image-based indoor positioning system: fast image matching using omnidirectional panoramic images," in *Proceedings of the 1st ACM International Workshop on Multimodal Pervasive Video Analysis*, MPVA '10, (Firenze, Italy), pp. 1–4, ACM, October 2010.

[21] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proceedings of the 7th IEEE/ACM International*

Symposium on Mixed and Augmented Reality, ISMAR '08, (Cambridge, UK), pp. 125–134, IEEE, September 2008.

[22] M. Schellenbach, A. Krüger, M. Lövdén, and U. Lindenberger, "A laboratory evaluation framework for pedestrian navigation devices," in *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*, Mobility '07, (Singapore, Malaysia), pp. 495–502, ACM, September 2007.

[23] W. Wahlster, J. Baus, C. Kray, and A. Krüger, "REAL: Ein ressourcenadaptierendes mobiles Navigationssystem," *Informatik-Forschung und Entwicklung*, vol. 16, no. 4, pp. 233–241, 2001.

[24] A. Krüger, A. Butz, C. Müller, C. Stahl, R. Wasinger, K.-E. Steinberg, and A. Dirschl, "The connected user interface: Realizing a personal situated navigation service," in *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, (Funchal, Madeira, Portugal), pp. 161–168, ACM, January 2004.

[25] T. Höllerer, S. Feiner, D. Hallaway, B. Bell, M. Lanzagorta, D. Brown, S. Julier, Y. Baillot, and L. Rosenblum, "User interface management techniques for collaborative mobile augmented reality," *Computers & Graphics*, vol. 25, no. 5, pp. 799–810, 2001.

[26] A. Möller, M. Kranz, R. Huitl, S. Diewald, and L. Roalter, "A mobile indoor navigation system interface adapted to vision-based localization," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, (Ulm, Germany), pp. 1–10, ACM, December 2012.

[27] A. May, T. Ross, S. Bayer, and M. Tarkiainen, "Pedestrian navigation aids: information requirements and design implications," *Personal and Ubiquitous Computing*, vol. 7, no. 6, pp. 331–338, 2003.

[28] E. Sadalla, W. Burroughs, and L. Staplin, "Reference points in spatial cognition," *Journal of experimental psychology: Human learning & memory*, vol. 6, no. 5, pp. 516–528, 1980.

[29] M. Raubal and S. Winter, "Enriching wayfinding instructions with local landmarks," in *Geographic Information Science* (M. Egenhofer and D. Mark, eds.), vol. 2478 of *Lecture Notes in Computer Science*, pp. 243–259, Springer Berlin Heidelberg, 2002.

[30] K. Lovelace, M. Hegarty, and D. Montello, "Elements of good route directions in familiar and unfamiliar environments," in *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science* (C. Freksa and D. Mark, eds.), vol. 1661 of *Lecture Notes in Computer Science*, pp. 65–82, Springer Berlin Heidelberg, 1999.

[31] M. Sorrows and S. Hirtle, "The nature of landmarks for real and electronic spaces," in *Spatial*

Information Theory. Cognitive and Computational Foundations of Geographic Information Science (C. Freksa and D. Mark, eds.), vol. 1661 of *Lecture Notes in Computer Science*, pp. 37–50, Springer Berlin Heidelberg, 1999.

[32] B. Elias, "Erweiterung von Wegbeschreibungen um Landmarks," *Publikationen der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung*, vol. 11, pp. 125–132, 2002.

[33] B. Elias and M. Sester, "Landmarks für Routenbeschreibungen," *GI-Technologien für Verkehr und Logistik. Beiträge zu den Münsteraner GI-Tagen 20./21. Juni 2002*, vol. 13, pp. 383–402, 2002. Institut für Geoinformation, Münster 2002.

[34] C. G. Healey and J. T. Enns, "Attention and visual memory in visualization and computer graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 7, pp. 1170–1188, 2012.

[35] B. Brunner-Friedrich and V. Radoczky, "Active landmarks in indoor environments," in *Visual Information and Information Systems* (S. Bres and R. Laurini, eds.), vol. 3736 of *Lecture Notes in Computer Science*, pp. 203–215, Springer Berlin Heidelberg, 2006.

[36] H. Hile, R. Grzeszczuk, A. Liu, R. Vedantham, J. Košecka, and G. Borriello, "Landmark-based pedestrian navigation with enhanced spatial reasoning," in *Pervasive Computing* (H. Tokuda, M. Beigl, A. Friday, A. Brush, and Y. Tobe, eds.), vol. 5538 of *Lecture Notes in Computer Science*, pp. 59–76, Springer Berlin Heidelberg, 2009.

[37] C. Kray, C. Elting, K. Laakso, and V. Coors, "Presenting route instructions on mobile devices," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, (Miami, FL, USA), pp. 117–124, ACM, January 2003.

[38] S. Münzer, H. D. Zimmer, M. Schwalm, J. Baus, and I. Aslan, "Computer-assisted navigation and the acquisition of route and survey knowledge," *Journal of Environmental Psychology*, vol. 26, no. 4, pp. 300–308, 2006.

[39] G. Gartner and V. Radoczky, "Schematic vs. topographic maps in pedestrian navigation: How much map detail is necessary to support wayfinding," in *AAAI 2005 Spring Symposia*, (Pittsburgh, PA, USA), March 2005.

[40] K. Hornbæk, B. B. Bederson, and C. Plaisant, "Navigation patterns and usability of zoomable user interfaces with and without an overview," *ACM Transactions on Computer-Human Interaction*, vol. 9, no. 4, pp. 362–389, 2002.

[41] I. Aslan and A. Krüger, "The Bum Bag Navigator (BBN): An advanced pedestrian navigation system," in *UbiComp Workshop on Artificial Intelligence in Mobile Systems*, AIMS '04, (Nottingham, UK), pp. 15–19, September 2004.

[42] E. S. Bhasker, S. W. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate,

low-maintenance geolocationing," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, PERCOM '04, (Orlando, FL, USA), pp. 111–120, IEEE, March 2004.

[43] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A mobile context-aware tour guide," *Wireless networks*, vol. 3, no. 5, pp. 421–433, 1997.

[44] S. Diewald, L. Roalter, A. Möller, and M. Kranz, "Towards a Holistic Approach for Mobile Application Development in Intelligent Environments," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, MUM '11, (Beijing, China), pp. 73–80, ACM, December 2011.

[45] A. Möller, C. Kray, L. Roalter, S. Diewald, R. Huitl, and M. Kranz, "Tool Support for Prototyping Interfaces for Vision-Based Indoor Navigation," in *Proceedings of the Workshop on Mobile Vision and HCI (MobiVis). Held in Conjunction with Mobile HCI*, MobiVis '12, (San Francisco, CA, USA), September 2012.

[46] J. F. Kelley, "An iterative design methodology for user-friendly natural language office information applications," *ACM Transactions on Information Systems*, vol. 2, pp. 26–41, January 1984.

[47] O. Soulard, "Development of an adaptive user interface for mobile indoor navigation," Master's thesis, Lehrstuhl für Medientechnik, Fachgebiet Verteilte Multimodale Informationsverarbeitung, Technische Universität München, Sept. 2012.

[48] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *19th IEEE International Conference on Image Processing*, ICIP '12, (Orlando, FL, USA), pp. 1773–1776, IEEE, September 2012.

[49] M. W. Van Someren, Y. F. Barnard, J. A. Sandberg, *et al.*, *The think aloud method: A practical guide to modelling cognitive processes*. Academic Press London, 1994.