Spontaneous and Privacy-friendly Mobile Indoor Routing and Navigation

Peter Ruppel and Florian Gschwandtner

Mobile and Distributed Systems Group
Institute for Computer Science
Ludwig-Maximilians-Universität, Munich, Germany

peter.ruppel@ifi.lmu.de
florian.gschwandtner@ifi.lmu.de

Abstract: Mobile indoor navigation systems guide users through buildings and premises by taking into account the current position of the user and a destination such as a room, office or shop. In many environments a spontaneous and also anonymous navigation is desirable e.g. at airports, hospitals or university campuses. However, many systems that have been proposed so far require the integration of a radio-based positioning system and access to other organization-specific systems, which results in a lot of things a user has to deal with before the navigation system can be used and which raises a lot of issues with respect to the location privacy of the user. In this paper we propose a mobile indoor navigation system that allows spontaneous navigation and that guarantees location privacy for the user. A pre-compiled compact representation for all the routes and navigation instructions in a building is the basis for an application that uses the built-in camera on the mobile device to derive the current position in the building from barcodes which are distributed all over the building. We show that the approach allows to navigate through a building with thousands of rooms and requires only little data on the device. Routes can be computed efficiently in $O(1)$ on the device and the only infrastructure that is required are the barcodes which can be easily printed and installed at walls or already existing signs in the building.

1 Indoor Navigation

Guiding people through buildings by a navigation system involves the measurement of the users current position, the calculation of the shortest possible path to the destination and intelligible visual, audible or tangible means that tell the user where to walk to the destination. There are three major challenges for indoor navigation applications: First, the common concepts from outdoor road navigation applications cannot be directly applied as people are not walking on roads but can move freely through rooms, floors and large halls. Secondly, very accurate position and orientation information is required in order to give precise instructions. Current satellite positioning systems are far from providing an accurate position throughout large buildings and indoor positioning systems such as WiFi positioning involves a lot of infrastructure components and effort to set up the system.

Thirdly, the location privacy of the user not only needs to be protected but guaranteed within such systems. We are deeply convinced that even the most powerful indoor navigation system must ensure that the user has the total and exclusive control over his location data.

It is important to differentiate between *indoor maps applications* and *indoor navigation applications*. The former only display maps and positions and come without information about the topological structure of a building. The latter allow in addition the calculation of shortest possible paths and walking distances in the building together with precise instructions such as "walk straight ahead the corridor and after 20 meters turn right". Thereto indoor navigation requires an appropriate model of the building that comprises e.g. room polygons, graphs that reflect the accessibility and distances between rooms, or points of interest (POIs) and landmarks. Hence routing between arbitrary locations in the building is a required base mechanism for future indoor navigation applications.

Depending on the environment and the requirements of the navigation application a navigation instruction itself can basically be realized both on mobile devices and on stationary components that are installed in a building such as public displays, speakers, illuminated markers or digital signposts. Present mobile devices possess adequate displays and computing power to run navigation applications directly on the device. Together with existing and upcoming software platforms for purchasing, installing and maintaining the application it is easy to offer mobile navigation applications for the mass-market. However, the required means for indoor positioning raise several problems.

Suppose a user enters a building for the first time and just wants to get directed to a certain room without revealing his location to the owner of the building or to any other party. Such a situation is very common e.g. at university campuses, airports or hospitals. It requires an autonomous application running on the user's mobile terminal that is capable of providing the required location data as well as the information that is needed for the computation of navigation instructions. Unlike outdoor GPS, obtaining the location data from a local indoor positioning system usually involves a lot of manual input such as establishing a WiFi link to a specific network, entering user credentials or configuring some other positioning client on the device. This impedes a spontaneous use of the navigation system and at the same time network-based positioning (see below) can comprise the users privacy.

This paper presents a new approach for a mobile indoor navigation system that allows a spontaneous use and provides absolute privacy for the user. The main idea is to have an application running on a mobile device that holds all needed data internally and does not need to connect to a central server. A user can determine his current position in the building by utilizing the built-in camera of his mobile terminal and barcodes that are spread across the floors and rooms. For a survey on different possible barcodes see e.g. [KT07]. The mobile terminal holds a very compact data set for each building and can deduce routes, maps and navigation instructions in $O(1)$ from the data set. The remainder of this paper is structured as follows: below a classification of indoor navigation systems is presented. Afterwards navigation instructions and the modeling of building is discussed. In section 2.3 we show how pre-compiled routing information can be stored in in a very compact representation and how navigation instructions can be efficiently derived on the device. Section 3 concludes the paper and highlights future work.

1.1 Classification of indoor navigation systems

In general indoor navigation systems can be classified among seven different properties: the mobility, the degree of personalization, the instruction granularity, the positioning system, the location privacy, the route calculation process and dynamic behavior. Each property has different characteristics and affects both functional and non-functional qualities of the system.

1. *Mobility*: an indoor navigation application can be executed on a mobile device or on stationary components that are available in the building. Such stationary components include e.g. displays, lights, speakers or other input and output systems.

2. *Degree of personalization*: in some environments it is desirable to navigate all users to the same destination. At other locations such as airports instructions may be the same for a certain group, e.g. all users departing on the same flight. Personal navigation systems provide individual routes and directions and take into account the users personal preferences.

3. *Instruction granularity*: the required accuracy and complexity of navigation instructions may vary between different scenarios. The more fine grained they are the more complex the model of the building needs to be and the higher the computational complexity for route and directions calculation is.

4. *Positioning system*: numerous indoor positioning systems exists, ranging from positions the user can enter by herself, systems based on Radio Frequency Identification (RFID), Near Field Communication (NFC), infrared light, ultra sonic, Wifi, or Ultra Wide Band (UWB) communication. For indoor navigation purposes it is important how fast and accurate positions can be computed and – in case mobile terminals are involved – how much energy is needed on the device.

5. *Route calculation process*: computing shortest paths and finding optimal routes through a building can be a time-consuming process. especially on mobile devices when the routes need to be calculated frequently. Also it is easy nowadays to save the road network for a whole country on a single memory storage card but it would not (yet) be possible to save the routing information for each single building in that country on the same card.

6. *Dynamic behavior*: shortest possible paths may vary over time due to changes in the accessibility of rooms or e.g. the length of waiting queues at the security check inside an airport. Whether or not the navigation application can react to these changes depends on the availability of the context information and dynamic route calculation algorithms.

7. *Location privacy*: at all times the location of a user must not be revealed to anybody else nor any other system unless the user explicitly agrees to it. That is for providing a good location privacy a terminal-based positioning system is always preferable to a terminal-assisted or network-based positioning system. Thereby terminal-based

constitutes that both the physical measurements and the position calculation is done exclusively on the terminal, e.g. as with the Global Positioning System. On the other hand terminal-assisted and network based positioning systems compute positions on components that are normally not under the direct control of the user. The other thing that affects the location privacy is the way the navigation application handles location information.

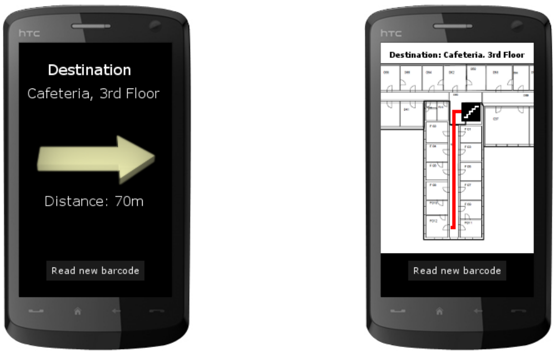
There are already a couple of existing indoor maps and navigation systems both with stationary and mobile devices. In [MWBS09] a client-server-based system is described that utilizes 2D barcodes and shows the position of the user and the destination on a map. The barcode recognition algorithm also identifies the local movement of the user with respect to the position of the barcode. Another approach is described in [CTW08], which guides people with cognitive impairments through buildings. The system also requires a server for computing the navigation instructions and involves an additional component to track the users.

In [CM02] the authors examine how various multi-modal representations of route descriptions affect the user navigation performance. The authors have selected six different information representations for guiding a navigation task. The first three variants only describe the next step which is required to reach the navigation goal. This is done by displaying an arrow to the direction of the next step or by using a single text command such as ”left” to describe that direction or by using audio commands to read the text command. In addition to these three approaches, more complex variants which describe a bigger part of the whole navigation task are evaluated: A text list with the next seven required steps, a map which shows the current location beacon and finally a full map displaying the current location beacon and the complete navigation environment including the entire route. An experimental evaluation shows that both an audio-based representation of route descriptions and the use of simple arrows lead to a significant decrease of time needed to complete the navigation task. Other systems such as the one described in [KKK05] use wireless connected stationary displays to dynamically route groups of people to their destination.

2 Spontaneous and Privacy-friendly Mobile Indoor Navigation

When people are walking through a building, navigation instructions are needed only sporadically to check up the next direction. In that moment the user gets displayed a simple arrow denoting the next direction together with the distance to the next way point. In addition the instructions can be enriched by a map that shows the whole or a part of the environment and the exact path to the destination. During this process a very short response time is desirable for the user. However, existing approaches that utilize a central server for instruction calculation are often subject to a delay in the order of magnitude of a couple of seconds, caused by the WiFi or 3G connection and the processing on the server.

We aim for an approach that 1) minimizes response time such that the user can just point his device to a barcode at the wall and gets displayed a simple and intelligible instruction in less than one second and 2) guarantees absolute location privacy for the user as no location



(a) (b)

Figure 1: (a) Simple navigation instruction. (b) Map overview.

information is ever exchanged between the mobile terminal and a server.

In the following, first it is described what kind of instructions are necessary on the mobile device and how buildings can be modeled to derive the needed information for instruction generation. Then a compact data representation is presented that encapsulates the navigation information on the device. In 2.4 we demonstrate to which extend a growing number of rooms affects the size of the data store on the device.

2.1 Indoor navigation instructions

Plain and simple instructions are important so that they can be easily followed by the users. A minimum set of instructions to guide people through a building are four directions {*straight ahead, left, right, backwards*}, a flag that denotes to move {*stairs upwards, stairs downwards*} and the instruction to use an {*elevator to floor #x*}. Figure 1(a) shows an example telling the user to proceed 70m to the right and figure 1(b) depicts a map with a route that can be displayed in addition to enhance intelligibility. Every time the user passes a barcode he can point his device to the barcode and a navigation instruction is displayed. As the position and orientation of the barcode is known by the system it is also possible to determine the viewing angle of the user through the barcode recognition algorithm and thus the according instruction can be derived. This approach allows the user to position himself at every discrete barcode location in the building. The alignment of the barcodes in the building can be in any order but it makes sense to put them at the waypoints which are described below.

2.2 Modeling buildings

Existing approaches for describing the structure of a building can be divided into geometric, graph-based and hybrid models, where each of them has certain advantages and disadvantages [BD05, HL04]. Geometric models describe the shape of each room but do not cover accessibility. With a graph-based representation routes can be calculated but visibility of objects cannot be calculated due to missing information about the walls and obstacles. For intelligible navigation applications a hybrid model is required, which covers both shape and hierarchy for rooms.

Inside a building a waypoint represents a location where people who walk towards a destination usually would have to choose between different options for the next direction of motion. Waypoints can be e.g. at doors, stairs or at several locations in a hall. In the following, rooms, corridors, halls etc. are defined by concave or convex polygons and their connections to other rooms. A connection can be e.g. a door, an escalator, an elevator or a narrow passage. The waypoint graph $WG = (V, E)$ is a directed weighted graph where the set V of nodes represents the waypoints and the set E of edges denotes the possible paths a person can walk between the waypoints. Different approaches for generating the waypoint graph exist. The simple approach is to draw the polygons and graphs by hand but this is often not feasible for large premises. Automatic waypoint graph generation can be achieved e.g. by analyzing digital CAD files and refinement through algorithms such as the corner graph, convex partitioning, medial axis or Delaunay triangulation. Such algorithms partition concave into convex polygons. Given WG and the set of polygons it is easy to calculate the walking distance between two arbitrary locations in the building.

2.3 Pre-compiling routes

In the following, the comprehensive building model is utilized to extract a compact data store that can be loaded into the mobile navigation application. For each building that the user visits he will have to download a file containing the pre-calculated routes which are described below. Upon entering the building the user can select his desired destination and every time he points the built-in camera to a barcode on the wall he gets displayed the navigation instructions.

For a given barcode identifier (BID) and a destination the data store on the device delivers the accordant instruction together with different distance information. To maximize the speed of this process the data store contains pre-calculated routes. But as neither the starting position nor the destination are known during creation of the data store, it is necessary to calculate a route for every pair of barcodes (b_i, b_j) with $i \neq j$ which is known as the *all pairs shortest paths* problem. Therefore the use of an algorithm adapted for that problem like Floyd-Warshall algorithm or Johnson's algorithm is recommended.

The basis for the pre-calculation process are 1) information about maps for every floor associated with a certain pixel per meter ratio (PPM) and 2) information about every barcode in the building. A description for every barcode can be used to allow an easy

```
<DATA>              ::= <HEADER> <MAPS> <BAR_INFO> <BAR_DEST>
<HEADER>            ::= <VERSION> <NAME_BUILDING> <LENGTH_BARCODE> <LENGTH_FLOOR>
<MAPS>              ::= <LENGTH_MAPS> (<FID> <PPM> <LENGTH_MAP> <MAP>)*
<BAR_INFO>          ::= <LENGTH_INFO> (<BARCODE> <DESCR> <FID> <X> <Y>)*
<BAR_DEST>          ::= <LENGTH_DEST> (<BARCODE>, <LENGTH_BARDEST> (<BARCODE> <DIR> <LENGTH_SEGMENT> <BARCODE>)*)*
<BARCODE>           ::= Number of a barcode
<X>                 ::= X coordinate of a barcode on a map
<Y>                 ::= Y coordinate of a barcode on a map
<MAP>               ::= Image of a floor
<VERSION>           ::= Protocol version
<NAME_BUILDING>     ::= Name of building
<LENGTH_BARCODE>    ::= Number of bits used to represent a barcode
<LENGTH_FLOOR>      ::= Number of bits used to represent a floor
<LENGTH_MAPS>       ::= Number of bits used for all floor descriptions together
<FID>               ::= Floor ID
<PPM>               ::= Pixel per meter ratio
<LENGTH_MAP>        ::= Size (in bits) of a map
<LENGTH_INFO>       ::= Number of bits used for all barcode information
<DESCR>             ::= Description of a barcode
<LENGTH_DEST>       ::= Number of bits used for all routes
<LENGTH_BARDEST>    ::= Number of bits used for all routes starting from one specific barcode
<DIR>               ::= Direction to go to next barcode
<LENGTH_SEGMENT>    ::= Distance between current position and next barcode
```

Figure 2: Data format defined in Backus-Naur Form

search for destinations on the mobile terminal, as it can contain e.g. the name of a person sitting within an office tagged by that specific barcode. Therefore, the user can specify the destination just by selecting the person's name. To be able to draw barcodes on a floor image, it is also necessary to include the coordinates for every barcode, which are described by an (x, y)-coordinate and floor identifier.

Finally routing information needs to be stored. As a route from every barcode to every other barcode has to be considered and the primary goal is to save memory, only the first segment of every route is stored within the data store. For example: If a route from barcode b_a to barcode b_e ($b_a \rightarrow b_b \rightarrow b_c \rightarrow b_d \rightarrow b_e$) should be stored, actually only the first segment ($b_a \rightarrow b_b$) of that route is inserted into the data store. A user starting at b_a with the destination b_e first has to go to b_b. Since the same procedure is done for a route from b_b to b_e, the data store also holds that a user at b_b with the destination b_e has to go to b_c etc., so it is able to derive the full route just by appending route segments. Therefore, for every pair of barcodes, only one tripel has to be inserted into the data store, containing the next barcode and both a direction and a distance to that barcode. That way it is possible to derive a navigation instruction in $O(1)$ because only one lookup is necessary to fetch the information which is needed to guide the user on his next path segment. In the example given above the whole route can be retrieved through four lookup operations.

The data store is saved directly within a file using a binary data format, depicted in Backus-Naur Form (BNF) in Figure 2. This data format tries to minimize the memory space that is required to save all relations mentioned before by avoiding repeating data entries and by using a binary representation as this can further decrease the memory needed for each entry. The header is used to define the length of barcode and building fields, which enables the system to adjust entries to the minimum size required to uniquely identify every room

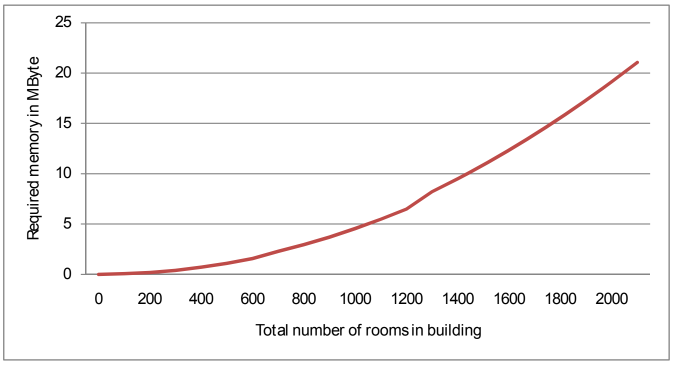


Figure 3: Memory usage of presented data format as a function of total number of barcodes

on every floor. Since all entries in $MAPS$, BAR_INFO, and BAR_DEST have either a fixed or a declared size (e.g. $LENGTH\_MAPS$), it is possible to skip irrelevant entries to speed up search.

2.4 Memory usage for pre-compiled routes

The described data format is able to combine all information required to navigate a user through a building with the help of barcodes. Since the primary goal of this data format is to minimize required memory, it is important that it is applicable for all kinds of buildings.

Let n be the total number of barcodes that are deployed in a building. In many environments this will be a number equal to the total number of rooms in the building. Let f be the total number of floors in the building. The total memory MEM that is needed to store the pre-compiled routes and barcode information for one building can be computed according to the following Equation 1:

$$\begin{aligned} MEM = {} & (Length_{BID} * n) * (Length_{Desc} + Length_X + Length_Y) \\ & + n * (n-1) * (3 * Length_{BID} + Length_{Dir}) \\ & + overhead \end{aligned} \tag{1}$$

where $Length_{Desc}$ is the maximum bit length of a room description, $Length_X$ and $Length_Y$ are the bit length of the coordinates and $Length_{BID}$ is the bit length of a barcode identifier which is defined by the following equation:

$$Length_{BID} = \left\lceil \log_2(\frac{n}{f}) \right\rceil + \lceil \log_2(f) \rceil = \lceil \log_2(n) \rceil \tag{2}$$

The equation 1 is plotted in Figure 3 as a function of total number of barcodes. For that we used a fixed length of 20 characters (equals 160 bit) for room descriptions and 2 bytes (equals 16 bit) for each coordinate. It shows that 2000 rooms, saved in the presented data format, use less than 20 Megabytes of memory. Therefore, an average building can be easily stored on modern mobile phones.

3 Conclusion and Outlook

In this paper we have classified future indoor navigation systems and have described a new device-centric and privacy-friendly approach for mobile indoor navigation. Users can determine their current location by pointing the camera of their mobile phone to 2D barcodes that are distributed among the building. A compact data representation is presented which allows to store all required routing and navigation information on the device. Thus no position data is exchanged with any other server and the location privacy of the user is guaranteed. The proposed system enables spontaneous usage as no building-specific external positioning or communication system is needed. A growing number of rooms in the building considerably affects the size of the data store on the device because of the underlying all pairs shortest paths problem. However, the size is feasible for environments with thousands of rooms per building.

There are several problems to be considered in future work. The inertial sensors and the compass in mobile phones could be utilized to further improve the alignment of navigation instructions, even when the user is not standing in front of a barcode. Secondly, the presented approach requires a user to download one single file per building. It is also possible to improve the data store creation algorithm in order to allow dynamic reloading of building topology data in the background. Another idea is to incorporate other information for terminal-based positioning in the data store. Such information could include e.g. WiFi fingerprints or RFID tag identifiers. That way other positioning systems can be used too without compromising the location privacy of the user.

References

[BD05] Christian Becker and Frank Dürr. On location models for ubiquitous computing. *Personal Ubiquitous Comput.*, 9(1):20–31, 2005.

[CM02] C. M. Chewar and D. Scott McCrickard. Dynamic route descriptions: tradeoffs by usage goals and user characteristics. In *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*, pages 71–78, New York, NY, USA, 2002. ACM.

[CTW08] Yao-Jen Chang, Shih-Kai Tsai, and Tsen-Yung Wang. A context aware handheld wayfinding system for individuals with cognitive impairments. In *Assets '08: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 27–34, New York, NY, USA, 2008. ACM.

[HL04] Haibo Hu and Dik-Lun Lee. Semantic Location Modeling for Location Navigation in Mobile Environment. *IEEE International Conference on Mobile Data Management*, 0:52, 2004.

[KKK05] Christian Kray, Gerd Kortuem, and Antonio Krüger. Adaptive navigation support with public displays. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 326–328, New York, NY, USA, 2005. ACM.

[KT07] Hiroko Kato and Keng T. Tan. Pervasive 2D Barcodes for Camera Phone Applications. *IEEE Pervasive Computing*, 6(4):76–85, 2007.

[MWBS09] Alessandro Mulloni, Daniel Wagner, Istvan Barakonyi, and Dieter Schmalstieg. Indoor Positioning and Navigation with Camera Phones. *IEEE Pervasive Computing*, 8(2):22–31, 2009.