A Mobile RFID-based System for Supporting Evacuation of Buildings

Luca Chittaro and Daniele Nadalutti

HCI Lab
Dept. of Math and Computer Science
University of Udine
via delle Scienze, 206
33100 Udine, Italy
{chittaro, daniele.nadalutti}@dimi.uniud.it

Abstract. Natural and man-made disasters present the need to efficiently and effectively evacuate the people occupying the affected buildings. In such situations, people are usually under stress and the use of a location-aware mobile application for giving evacuation instructions in simple and effective ways can be useful to improve users' decision making. This paper proposes a mobile system that gives evacuation instructions by employing an interactive 3D location-aware model of the building. The main focus of the paper is on the solutions and the technologies adopted for determining user's position into the building and for interactively visualizing a 3D model of the building augmented with visual evacuation instructions on mobile devices.

1 Introduction

Natural and man-made disasters present the need to efficiently and effectively evacuate the people occupying the affected buildings. The occupants have to be evacuated as soon as possible (e.g., in case of fire) or immediately after the event (e.g., in case of earthquake). Considering the complexity of large buildings and the possible large number of occupants, it is often difficult to organize a quick evacuation, especially when the building is seriously damaged [1]. Moreover, people in a disaster are usually under stress and may "freeze", leading to fatalities in otherwise survivable conditions [2]. Finally, in large public buildings like airports, the occupants can also be unaware of the topology of the building or the location of the emergency exits and they are usually not trained in evacuating such buildings.

A location-aware mobile application for giving evacuation instructions in simple and effective ways can be useful to improve users' decision making, preventing users' errors and minimizing casualties. Moreover, location-aware mobile applications can be used for training purposes, providing the user with emergency simulations, so that she can learn evacuation paths for different scenarios by actually following them in the building to gain knowledge and abilities that will be useful in real emergencies. Moreover, users' actions can be logged by the application for post-training analysis.

It must be noticed that distasters often cause power outages in the affected building. For this reason, the technologies (e.g., wireless networks) adopted by the mobile application (e.g., for positioning) should not require availability of electrical power in the building to work properly.

This paper proposes a mobile system that uses 3D models of the building for giving evacuation instructions to the user. The system employs a mobile 3D rendering engine [3] to interactively visualize a location-aware 3D model of the building augmented with visual evacuation instructions. User's position into the building is determined by using active short-range RFID technology without the need for an electrical network. The system supports also manual navigation of the model for training purposes or if automatic positioning is not available.

The paper is organized as follows. Section 2 will briefly discuss related work. Section 3 will describe our system, analyzing its components and motivating the major design choices. Section 4 will provide conclusions and outline future work directions.

2 Related Work

A location-aware mobile system to support occupants' evacuation needs to rely on an appropriate positioning technology and to present navigation instructions in an easy-to-understand way.

Several technologies can be employed for indoor positioning (e.g, Infrared, indoor GPS, RFID, UWB, GSM, WLAN, Bluetooth, UHF, Ultrasound). Liu et al. [4] present a survey of wireless indoor positioning systems. They compare performance of several approaches in terms of accuracy, precision, robustness, scalability, complexity and cost.

Two well-known indoor localization systems (SpotON [5] and LANDMARC [6]) are based on active RFID technology. Both systems track the position of a specified tag by measuring its distance from multiple RFID readers that are placed at specific locations. The distance between a tag and a reader is computed based on received signal strength. To increase accuracy without placing more RFID readers, the LANDMARC system also uses a set of RFID tags, called reference tags, that are placed at fixed locations and serve as reference point for the system. The position of the tracked tag is computed as the weighted average of the positions of the k nearest reference tags, where weighting factors are based on estimated distances between the tracked tag and the k reference tags. However, these solutions for indoor localization based on RFID technology are not suitable for mobile emergency applications because they need a network to allow the communication between each RFID reader and a server where the position of the tracked tag is computed. Moreover, a wireless network is also needed for sending the computed position back to the mobile device. An alternative approach, less accurate but more suited to emergency situations, consists in using a single mobile RFID reader (e.g., Compact Flash RFID reader) on the mobile device and a set of tags that are placed at fixed locations: the mobile device can autonomously compute its position based only on distance between the reader and the tags without the need for network infrastructure.

Presentation of navigation instructions on mobile devices is a widely discussed topic in the literature. Baus et al. [7] surveyed the different solutions employed in mobile guides, especially for tourists. Most existing solutions are based on 2D maps, but alternative approaches have been studied, such as photos of the environment augmented with visual navigation aids (e.g., arrows); 3D models; textual instructions; audio directions; route sketches. Approaches based on 2D maps have the advantage of exploiting a well-known method for representing spatial information, but 3D models or augmented photos exploit natural users' spatial abilities because they provide users with the same visual cues they exploit in the real world (e.g., occlusion, size of the objects). Moreover, solutions based on 3D models might allow the user to train in navigating a building without being in it. However, using 3D graphics on mobile devices for navigation purposes is currently a scarcely explored subject in the literature. The first investigations were thought for outdoor environments [8, 9]. Later, some projects explored the use of 3D models for helping users in the navigation of indoor environments [10]. Garcia Barbosa et al. [10] developed a framework which allows users to load 3D models from a remote PC server, navigate them, find an optimal and collision-free path from one place to another, and obtain additional information on objects. A significant limitation of the framework is the lack of automatic positioning: the user has to navigate the model manually. Moreover, the employed 3D models are very simple and this could make it difficult for users to visually match them with the real world. Finally, the framework needs a wireless network infrastructure to compute paths.

In recent years, only a few attempts have been made at exploring the use of 3D graphics on mobile devices for presenting evacuation instructions. Garcia Barbosa et al. [10] considered the application of their framework for virtual rescue training of firefighters. Pu and Zlatanova [1] list instead the requirements for a mobile system and a framework to manage evacuation of buildings using 3D models, but they do not implement it.

3 The proposed system

To the best of our knowledge, our system is the first mobile system that uses location-aware 3D models of buildings for evacuation purposes. The system represents paths by means of a set of bidimensional oriented arrows that are projected on the floor. Emergency exits are highlighted by using spotlights [11] (Figure 1).

The system uses a single compact flash RFID reader on the mobile device and a set of tags placed at fixed locations to determine user's position in the building and to consequently update the position and the orientation of the viewpoint in the 3D model. As it is typical of navigators, there are some limitations in determining user's orientation. The system, indeed, computes user's

Fig. 1. A 3D model augmented with evacuation instructions. Paths are represented by arrows (left figure), while landmarks are highlighted by spotlights (right figure).

orientation from her latest positions, so if the user makes a turn without significantly changing her position in the world, the system is not able to recognize the change of orientation. Inaccurate computed orientations can lead to wrong and useless viewpoints in the 3D model (e.g., a viewpoint that is very close to a wall and orientated towards the wall), with consequent difficulties for users to match their position in the real world with the 3D world. To avoid this, we adopt a solution inspired by car navigators and we snap user's positions and orientations to the evacuation path.

3.1 Architecture and functions

Figure 2 illustrates the architecture of the proposed system, composed by three main modules: the *Viewpoint Calculator*, the *Path Planner*, and the *MobiX3D Viewer*.

The Viewpoint Calculator reads the queue of detected tags from the RFID reader, retrieves their coordinates in the real world from the Tag Positions database, and then computes the user's current position and her orientation. Finally, it sends the corresponding position and orientation of the viewpoint in the 3D model to the Path Planner. We use a reader for Beacon RFID tags, i.e. active tags that periodically send a signal to the reader. The specific tags we use have a range of about 4 meters and send their signal to the reader every 500 milliseconds. The RFID reader simply stores the detected tags in a queue that is queried by the Viewpoint Calculator every 500 milliseconds.

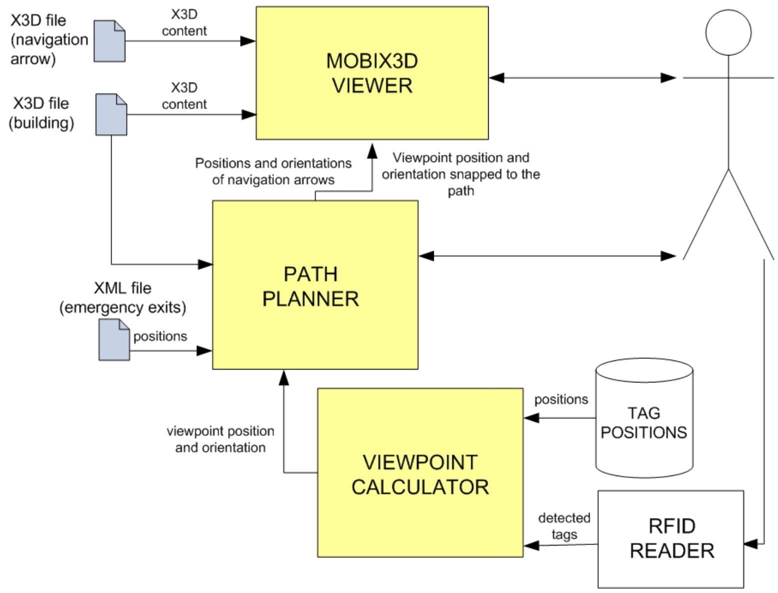

Fig. 2. Architecture of the proposed system

The Viewpoint Calculator then computes position and orientation of the viewpoint in four steps: (i) computation of a rough viewpoint position (i.e., a position derived by triangulating the detected RFID tags), (ii) computation of current viewpoint position by filtering the latest rough viewpoint positions, (iii) computation of rough viewpoint orientation from the latest two current viewpoint positions, (iv) computation of current viewpoint orientation by filtering the latest rough viewpoint orientations.

In the first step, the Viewpoint Calculator computes a rough viewpoint position by using a simple algorithm based on detected tags and on the strength of their signal. For each detected tag, the algorithm estimates the distance d from the tag based on signal strength (which decreases exponentially as one moves away from the tag) and is computed using the following formula [12]:

$$d = 10^{\frac{(P_0 - P(d))}{10n}}$$

where P_0 is the signal strength at 1 m, $P(d)$ is the signal strength at distance d, n is a constant that determines how the signal strength decreases as the distance increases and has to be tuned empirically. In our tests, n has be tuned to the behavior of the specific tags and reader we employ (and is equal to 2.4967854). However, if the signal strength associated to a tag is over a certain threshold, which corresponds to the typical strength obtained when the reader is very close (< 50 cm) to a tag, then we set d equal to 0.5 m. The rough viewpoint position is computed by using triangulation. When no tags are detected, the rough viewpoint position that is generated is the one computed 500 milliseconds before. For this reason, the computed viewpoint in the 3D model might at times suffer from slight delays in update. In extreme cases, if no tags are detected for a minute, the system warns that the viewpoint in the 3D model could not be in sync with the actual position of the user.

In the second step, the Viewpoint Calculator computes the current viewpoint position, i.e. the position sent to the Path Planner, by filtering the latest rough viewpoint positions. We currently use a simple filter that computes the mean of the latest 5 rough viewpoint positions. In the first 2 seconds, when less than 5 rough viewpoint positions are available, the mean is computed considering the available rough viewpoint positions.

In the third step, the Viewpoint Calculator computes a rough viewpoint orientation as the vector between the latest two viewpoint positions.

In the fourth step, the Viewpoint Calculator computes the current viewpoint orientation, i.e. the orientation sent to the MobiX3D viewer, by filtering the latest rough viewpoint orientations in the same way of the positions. The Viewpoint Calculator sends current viewpoint position and orientation to the Path Planner every 500 milliseconds.

The Path Planner has two main functions: (i) computing the evacuation path from current position to the nearest emergency exit, and (ii) snapping position and orientation of the viewpoint computed by the Viewpoint Calculator into the current path. To compute evacuation paths, the Path Planner uses current viewpoint position and orientation, a 2D map of the building (derived from the 3D model) and the position of the emergency exits. The evacuation path is represented as a directed acyclic graph G where each node is associated to a waypoint and two consecutive waypoints are connected by an edge. Formally, $G = (V, E)$, where $V = \{v_0, \ldots, v_{n-1}\}$, $v_i = (x_i, y_i, z_i)$ is a point in space, v_0 is located at the current position, and v_{n-1} is located at the nearest emergency exit. E is defined as follows:

$$E = \{(v_i, v_j) | v_i, v_j \in V, i = \{0, \ldots, n-2\}, j = i + 1\}$$

For each edge, the Path Planner sends to the MobiX3D Viewer the position and the orientation of a navigation arrow. The position of the i-th navigation arrow is the mean of of v_i and v_{i+1}, while its orientation is the unit vector pointing from v_i towards v_{i+1}.
To snap the position and the orientation of the viewpoint to the evacuation path, the Path Planner locates the node v_j of the evacuation path nearest to the viewpoint and sends the coordinates of v_j as viewpoint position and the unit vector pointing from v_j towards v_{j+1} as viewpoint orientation to the MobiX3D Viewer.

The MobiX3D Viewer displays the 3D model of the building augmented with the evacuation arrows. It also allows the user to switch among automatic and manual navigation modes. Automatic navigation mode updates the viewpoint in the 3D model based on viewpoint positions and orientations sent by the Path

Planner. Manual navigation mode allows the user to navigate the model by pressing the cursor keys of the mobile device. It is useful for training purposes or if no RFID tags are available. The input of the MobiX3D Viewer is the model of the building, the position and the orientation of the viewpoint in the 3D world, the model of the navigation arrows, their position and their orientation. The MobiX3D Viewer was originally proposed in [3] as a general X3D file viewer and was later refined [13] with a basic view frustum culling algorithm and extended with a portal culling algorithm [14] for buildings. The portal culling extension is used for very large buildings, when the entire 3D model cannot be loaded in memory. To test the evacuation system, we used a model of our Department, made of 50.000 triangles. The size of the source file is 4.38 MB, with 100 kB of textures, and can be loaded in memory without using the portal culling extension.

3.2 Tests on tag availability

We tested the system positioning algorithm on three different tag setups: (i) 4 tags placed about 8 m away from each other in a 24 meter corridor, (ii) 4 tags placed on the vertices of a 4-meter square, and (iii) 9 tags placed on a 4-meter square, following a regular 3x3 grid pattern (the 4-meter square was divided into four 2-meters squares).

We performed a walk along the corridor at a constant speed in the first setup, and a walk into the square following random trajectories at a constant speed in the second and the third setups. No other people were in the areas where tests were performed. The system logged the number of detected tags each time it sent a rough viewpoint position to the filter (i.e., every 500 milliseconds).

In the first setup, no tags were detected 49% of times when a rough viewpoint position was sent to the filter, 1 tag was detected 42% of times, and more than 1 tag was detected 9% of times. In the second setup, no tags were detected 7% of times, 1 tag was detected 46% of times, and more than one tag was detected 47% of times. Finally, in the third setup, no tags were detected 3% of times, 1 tag was detected 6% of times, and more than 1 tag was detected 91% of times.

Although the percentage of tag detection in the first setup was not high, it caused only an intermittent and slight delay in viewpoint updating, and one could easily match the current position in the real world with the position of the viewpoint in the 3D model. However, the second and the third setups guaranteed a higher refresh rate of the viewpoint, allowing for more immediate matching of movements in the real world with viewpoint changes.

3.3 Accuracy and precision

We preliminary measured the accuracy and the precision of our positioning algorithm by following the methodology described in [15]. We used the metric error distance, i.e., the spatial distance between the real position and the position computed by our positioning algorithm.

We carried out two tests. In the first one, we employed the third tag setup described in Section 3.2. We randomly chose a number of positions in the square and placed the mobile device equipped with the RFID reader at such positions. The mobile device remained in each position for 15 seconds (30 current viewpoint positions were computed) before moving to another position. The accuracy in terms of mean distance error was 0.8 m, with a precision of 90% within 1.5 m.

In the second test, we employed the first tag setup described in Section 3.2. We walked along the corridor at a constant speed for a number of times and the walking speed varied among the different walks. We measured an accuracy of 2 m in terms of mean distance error, with a precision of 75% within 3 m.

3.4 Placing tags in the building

Placement of tags in the building is a crucial aspect for the accuracy of positioning. An optimal placement guarantees the full coverage of the building by using the minimum number of tags without losing accuracy. This problem is considered by the EasyReader visual tool [16] for placing RFID tags, antennas and interrogators. The tool provides the user with a 2D visual map of the floors of the building and allows the user to drag and drop RFID components into the 2D map and visually show the coverage. Once the user places the components in a satisfactory way, the tool automatically generates a bill of materials for deployment and installation. However, that tool is aimed at designing configurations of RFID readers placed at fixed locations and used to track the position of moving RFID tags.

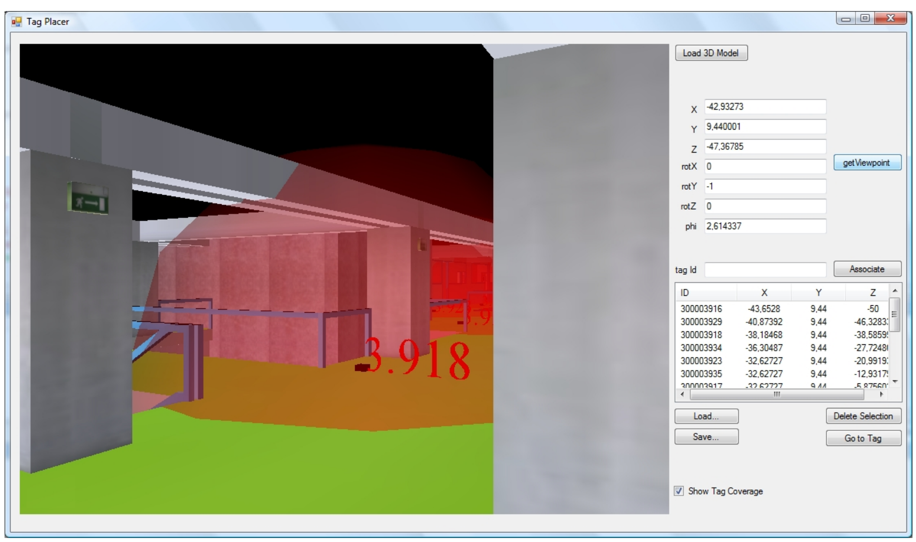

Fig. 3. Screenshot of our visual tool for placing tags. Each semitransparent sphere represents the coverage of the RFID tag.

We instead developed a visual tool, called TagPlacer that: (i) is aimed at configurations of tags (to be placed at fixed locations) that will be read by a single mobile RFID reader, and (ii) employs 3D models of buildings to facilitate the design. The tool helps the designer to quickly place tags in the building (Figure 3) by navigating the model and placing tags around. The tags are represented by black boxes labeled with red texts that indicate the tag ID. Moreover, spheres that represent the coverage of each tag can be displayed, so one can easily check whether there are uncovered areas or large overlaps among coverages of different tags in the part of building she would like to cover. Finally, one can directly store the associations between positions and tag IDs in a Tag Positions database (Figure 2) which is then used by our mobile application, and can load and edit other existing Tag Position databases.

4 Conclusions and Future Work

The system has been informally evaluated on 11 users in our Department with positive results that are described in a companion paper [17]. After focusing mainly on positioning and navigation support, our research is now proceeding in several directions. Firstly, we will improve the computation of the evacuation path, allowing the user to avoid the parts of the building that are damaged and unaccessible. This feature is crucial for evacuation purposes because the shortest path can cross areas of the building which have become dangerous. Some techniques have been proposed to efficiently compute the evacuation path with those additional constraints [1]. Then, we will improve the Tag Placer tool with automatic suggestion of optimal tag placements that cover the building. Algorithms for optimally placing RFID readers in traditional configurations with multiple RFID readers at fixed positions [18] can be a source of inspiration. Moreover, we will develop a mobile extension of the tool for helping the user in placing RFID tags on the field. Finally, we will consider the status of the battery of the mobile device to influence rendering accuracy and viewpoint updating frequency.

5 Acknowledgments

Our research has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under the PRIN 2005 project "Adaptive, Context-aware, Multimedia Guides on Mobile Devices".

References

[1] Pu, S., Zlatanova, S.: Evacuation route calculation of inner buildings. In van Oosterom, P., Zlatanova, S., Fendel, E.M., eds.: Geo-Information for Disaster Management, Berlin, Germany, Springer Verlag (2005) 1143–1161

[2] Leach, J.: Why people 'freeze' in an emergency: Temporal and cognitive constraints on survival responses. Aviation, Space, and Environmental Medicine **75**(6) (2004) 539–542

[3] Nadalutti, D., Chittaro, L., Buttussi, F.: Rendering of X3D Content on Mobile Devices with OpenGL ES. In: Web3D '06: Proceedings of the eleventh international conference on 3D web technology, New York, NY, USA, ACM Press (2006) 19–26

[4] Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **37**(6) (2007) 1067–1080

[5] Hightower, J., Want, R., Borriello, G.: SpotON: An indoor 3D location sensing technology based on RF signal strength. Univ. Washington, Seattle, Tech. Rep. UW CSE 2002–02–02 (2000)

[6] Ni, L.M., Liu, Y., Lau, Y.C., Patil, A.P.: LANDMARC: Indoor location sensing using active RFID. Wireless Networks **10**(6) (2004) 701–710

[7] Baus, J., Chevers, K., Kray, C.: A survey of map-based mobile guides. In Meng, L., Zipf, A., Reichenbacher, T., eds.: Map-based mobile services – Theories, Methods and Implementations, Berlin, Germany, Springer Verlag (2005) 197–216

[8] Rakkolainen, I., Vainio, T.: A 3D City Info for mobile users. Computers & Graphics **25**(4) (2001) 619–625

[9] Laakso, K., Gjesdal, O., Sulebak, J.: Tourist information and navigation support by using 3D maps displayed on mobile devices. In: Proceedings of Mobile HCI Workshop on HCI in Mobile Guides. (2003) 34–39

[10] Garcia Barbosa, R., Formico Rodrigues, M.A.: Supporting guided navigation in mobile virtual environments. In: VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology, New York, NY, USA, ACM (2006) 220–226

[11] Brusilovsky, P.: Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. PsychNology Journal **2**(1) (2004) 7–23

[12] Hallberg, J., Nilsson, M.: Positioning with Bluetooth,IrDA and RFID. Lulea University of Technology, MSc. Thesis (2002)

[13] HCI Lab – University of Udine: MobiX3D website. http://hcilab.uniud.it/MobiX3D (2006)

[14] Mulloni, A., Nadalutti, D., Chittaro, L.: Interactive walkthrough of large 3d models of buildings on mobile devices. In: Web3D '07: Proceedings of the twelfth international conference on 3D web technology, New York, NY, USA, ACM Press (2007) 17–25

[15] Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J., Gao, X.: A wireless LAN based indoor positioning technology. IBM Journal of Research and Development **48**(5) (2004) 617–626

[16] ODIN Technologies, Inc.: EasyReader. http://www.odintechnologies.com/index.php/software/easyreader (2006)

[17] Chittaro, L., Nadalutti, D.: Presenting evacuation instructions on mobile devices by means of location-aware 3d virtual environments. In: MobileHCI '08: Proceedings of the 10th conference on Human-computer interaction with mobile devices and services, New York, NY, USA, ACM Press (2008)

[18] Wang, L., Norman, B., Rajgopal, J.: Placement of multiple RFID reader antennas to maximise portal read accuracy. International Journal of Radio Frequency Identification Technology and Applications **1**(3) (2007) 260–277