Using Intelligent Mobile Devices for Indoor Wireless Location Tracking, Navigation, and Mobile Augmented Reality

Chi-Chung Alan Lo*, Tsung-Ching Lin*, You-Chiun Wang*, Yu-Chee Tseng,*
Lee-Chun Ko†, and Lun-Chia Kuo†
*Department of Computer Science, National Chiao-Tung University, Hsin-Chu 300, Taiwan
†Industrial Technology Research Institute, Hsin-Chu 310, Taiwan
Email: {ccluo, cclin, wangyc, yctseng}@cs.nctu.edu.tw; {brentko, jeremykuo}@itri.org.tw

***Abstract*—This paper proposes a new application framework to adopt mobile augmented reality in indoor localization and navigation. We observe that there are two constraints in existing positioning systems. First, the signal drifting problem may mistake the positioning results but most of positioning systems do not adopt user feedback information to help correct these positioning errors. Second, traditional positioning systems usually consider a two-dimensional environment but an indoor building should be treated as a three-dimensional environment. To release the above constraints, our framework allows users to interact with the positioning system by the multi-touch screen and IMU sensors equipped in the mobile devices. In particular, the multi-touch screen allows users to identify their requirements while the IMU sensors will feedback users' moving behaviors such as their moving directions to help correct positioning errors. Besides, with the cameras equipped in the mobile devices, we can realize the mobile augmented reality to help navigate users in a more complex indoor environment and provide more interactive location-based service. A prototype system is also developed to verify the practicability of our application framework.**

Keywords: augmented reality, inertial measurement unit, location tracking, pervasive computing, positioning.

I. Introduction

The location-based service is regarded as a killer application in mobile networks. A key factor to its success is the accuracy of location estimation. For outdoor environments, the global positioning system (GPS) has provided a mature localization technique. For indoor environments, many localization techniques based on received wireless signals have been proposed, and they can be classified into five categories: AoA-based [1], ToA-based [2], TDoA-based [3], path-loss [4], and pattern-matching [5]–[7] techniques. We focus on the pattern-matching technique such as RADAR [5] since it is more resilient to the unpredictable signal fading effects and thus could provide higher accuracy for positioning results in more complex environments.

However, existing pattern-matching solutions may face three problems:

- The pattern-matching technique relies on a training phase to learn the patterns of received signal strengths at a set of predefined locations from base stations or access points. However, such a training phase usually requires a large amount of human power to collect these patterns, especially in a large-scale area. Recently, this problem has been addressed in [8]–[10].
- In an indoor environment, the signal drifting problem could seriously mistake the positioning results. However, the positioning system believes only the received signal strengths and is lack of user feedback to correct these errors. In fact, such errors can be corrected by allowing users to manually revise their positions or letting the system automatically update the positioning results according to users' behaviors (such as moving directions).
- Traditional positioning systems usually consider a two-dimensional (2D) environment. While this is suitable for outdoor environments, an indoor building typically contains multiple floors and thus it should be considered as a three-dimensional (3D) environment. In this case, how to navigate users in such a complex environment is a challenging issue.

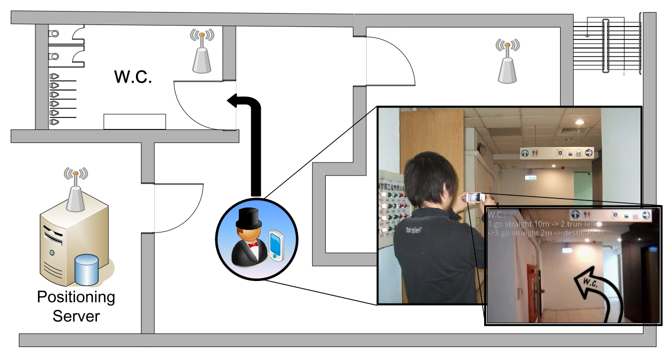


Fig. 1. Using AR on mobile devices to navigate users.

To address the last two problems of existing patten-matching systems, in this paper we propose using some intelligent mobile devices (e.g., smart phones) to realize location tracking and navigation in an indoor environment. Specifically, these mobile devices are equipped with *IMU (inertial measurement unit) sensors* to capture the moving behaviors of users (such as their current moving directions) and *multi-touch input interfaces* to let users interact with the positioning system. These components can feedback users' information to our positioning system to correct potential positioning errors caused by the signal drifting problem. In addition, we realize mobile *augmented reality (AR)* by the cameras equipped in the mobile devices to help navigate users to their destinations. The concept of our mobile AR is shown in Figure 1, where the user can take a picture from the environment and indicates his/her destination. Then, our positioning system will add some auxiliary descriptions (such as arrows and texts) on the picture to help navigate the user. Our contributions are two-fold. First, we integrate IMU sensors with intelligent mobile devices to feedback user's moving information to improve the accuracy of positioning results. Second, to the best of our knowledge, this is the first work that exploits mobile AR to help navigate users in a complex 3D environment. A prototyping system is demonstrated and the implementation experience is reported in this paper.

We organize the rest of this paper as follows. Section II gives some background knowledge. The design of our positioning system is presented in Section III. Section IV reports our implementation details. Conclusions and future work are drawn in Section V.

II. Preliminary

In this section, we give the background knowledge of pattern-matching localization, mobile AR, and IMU sensors.

A. Pattern-Matching Localization

A pattern-matching localization system usually consists of two phases, *training* and *positioning*. In the training phase, we are given a set of beacon sources $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ (from base stations or access points) and a set of training locations $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$. We then measure the *received signal strengths (RSS)* of the beacons generated from $\mathcal{B}$ at each training location $\ell_i \in \mathcal{L}$ to create a *feature vector* $\boldsymbol{\upsilon}_i = [\upsilon_{(i,1)}, \upsilon_{(i,2)}, \ldots, \upsilon_{(i,m)}]$ for ℓ_i, where $\upsilon_{(i,j)} \in \mathbb{R}$ is the average RSS of the beacon generated from b_j, $j = 1..m$. Then, the matrix $\mathcal{V} = [\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}_2, \ldots, \boldsymbol{\upsilon}_n]$ is called the *radio map* and used as the basis of positioning results.

In the positioning phase, a user will measure its RSS vector $\boldsymbol{s} = [s_1, s_2, \ldots, s_m]$ at the current location and compare s with each feature vector in $\mathcal{V}$. In particular, for each $\upsilon_{(i,j)} \in \mathcal{V}$, we define a distance function $h(\cdot)$ for the corresponding training location ℓ_i as [5]

$$h(\ell_i) = \|\boldsymbol{s}, \boldsymbol{\upsilon}_i\| = \sum_{j=1}^{n} \sqrt{(s_j - \upsilon_{i,j})^2}.$$

Then, the user is considered at location ℓ_i if $h(\ell_i)$ returns the smallest value.

B. Mobile Augmented Reality

To navigate users in a 3D environment, one possible solution is to adopt the technique of *virtual reality (VR)*, which constructs an "imaginary" world from the real world by computer simulations. VR is widely used in outdoor navigation systems. Figure 2 shows an example of GSP navigation system by using the VR technique. However, VR requires to pre-construct virtual maps for navigation, which could raise a very high cost for an indoor navigation system since we have to construct one virtual map for each floor.

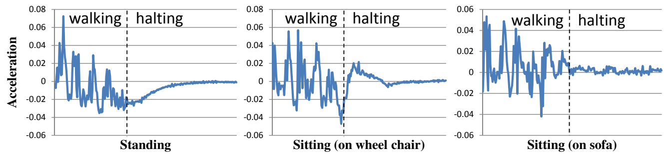
Fig. 2. A GPS navigation system using the VR technique.

On the other hand, AR is a variation of VR. Unlike VR, AR does not construct virtual maps but allows users to directly access the pictures of physical environments (which can be captured by cameras). Then, AR will add some computer-generated auxiliary descriptions such as arrows and texts on these pictures to help users realize the physical environment. By implementing AR on mobile devices, we can realize *mobile AR* to navigate users in an indoor environment, as shown in Figure 1. For example, the user can take a picture from the environment and indicates his/her destination. Then, the mobile device will add the description of each room on the picture and show an arrow to the destination. Due to its low construction cost and flexibility, we will implement mobile AR on intelligent mobile devices such as smart phones in this paper.

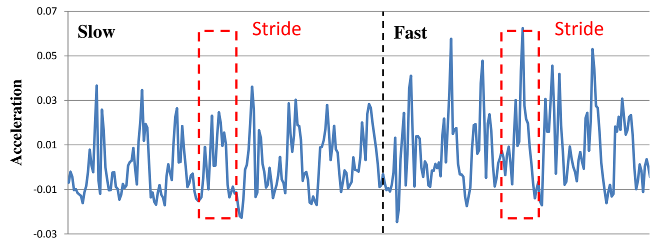
C. IMU Sensors

In an indoor environment, we are interested in four major moving behaviors of users: *halting*, *walking on the ground*, *going upstairs or downstairs*, and *taking elevators*. These moving behaviors can be tracked by IMU sensors such as tri-axial accelerometers (usually called *g-sensors*) and electronic compasses. A g-sensor can detect the 3D acceleration of a user and thus we can determine the mobility patterns of the user. For example, we can check whether this user is currently moving forward/backward/upstairs/downstairs or just halting. On the other hand, an electronic compasses can analyze the angle relative to the north and thus we can obtain the current (moving) direction of the user.

A g-sensor is quite sensitive to the user's mobility and thus we can easily analyze the moving behavior of that user. For example, Figure 3(a) shows the output signals when a user transits from a moving status to a halting status. We can



(a) when the user transits from a moving status to a halting status



(b) when the user is walking on the ground

Fig. 3. The output signals of g-sensors.

observe that when the user is halting, the output signal of the g-sensor is almost zero. On the other hand, Figure 3(b) shows the output signal of a g-sensor when the user is walking on the ground. We can also analyze whether or not this user is striding from the output signal.

III. System Design

Our objective is to develop a positioning system that not only allows users to interact with the system to correct their positioning results and identify their destinations, but also adopts mobile AR to navigate users to provide some location-based services such as answering "where is the restaurant?". To achieve this objective, we propose a system architecture as shown in Figure 4. In particular, each user carries a mobile device (such as a smart phone) equipped with a camera to take pictures from the environment and a multi-touch screen to interact with the positioning system. The indoor environment is deployed with WiFi access points that will periodically broadcast beacons. With the RSSs of the mobile device from these beacons, the positioning server can calculate the user's position. Then, based on the user's requirement (through the multi-touch screen), the system can add some auxiliary texts or arrows to realize mobile AR on the pictures of environment.

Our positioning system consists of two sides, *server side* and *mobile side*. The server side (executed on the positioning server) calculates users' locations and routing paths to their destinations, while the mobile side (executed on the mobile device) shows the positioning and navigation results by mobile AR. Each side contains four layers:

1) **Hardware layer:** This layer controls the hardware components of our positioning system.
2) **Positioning layer:** This layer calculates users' locations based on the RSS of the mobile devices and reports the positioning results to the navigation layer.
3) **Navigation layer:** According to the user's destination and current position, the navigation layer will estimate the shortest routing path between these two locations. Then, this path will be sent to the user-interface layer.
4) **User-interface layer:** This layer provides an interface for users to interact with the system and also supports some location-based services.

Below, we introduce the functionality of each component in the server side and mobile side.

A. Server Side

To avoid conducting too complicated computation at the mobile devices, we let most of calculations be handled on the server side. Then, the positioning server will send the calculation results to the mobile devices.

In the hardware layer, the server side requires a desktop computer or a laptop to conduct complicated calculations such as finding the user's position and estimating the shortest routing path between two given locations.

In the positioning layer, there are two components, *positioning pattern database* and *positioning engine*, used to calculate the positioning results. Here, we adopt the pattern-matching localization technique and thus the positioning pattern database stores the off-line training data. On the other hand, the positioning engine will obtain the RSS, user's mobility pattern, and user's feedback information such as the current moving direction from the mobile side. By comparing the RSS data and the training data from the positioning pattern database, the positioning engine can estimate the user's location. Besides, with the user's mobility pattern and feedback information, the potential positioning errors may be fixed. Then, the positioning engine will send the positioning result to the mobile side and the navigation layer.

The navigation layer has a *waypoint module* and a *routing module* to estimate the shortest routing path between two given locations. Our navigation idea is to select a set of waypoints from the indoor environment and then adopt any path-finding solution in graph theory to find out the shortest routing path. Specifically, we can select some locations along a pathway as waypoints. Crossways, stairs, and doors must be selected as waypoints. Then, we can construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model all possible paths, where the vertex set $\mathcal{V}$ contains all waypoints and the given source and destination while the edge set $\mathcal{E}$ includes all edges that connect any two adjacent, reachable vertices (i.e., a user can easily walk from one vertex to another vertex). All waypoints and their corresponding edges in $\mathcal{E}$ are stored in the waypoint module. Then, the routing module can exploit any path-finding scheme such as the Dijkstra's algorithm [11] to find the shortest routing path from the source (i.e., the current position of the user) to the destination.

Finally, the user-interface layer contains an *LBS (location-based service) database* and an *LBS module* to support some location-aware services such as "which place sells the food?" and "where is the nearest toilet?". The LBS database stores the service data (which could be supported by any third party) and can be replaced depending on the requirements. On the other hand, these service data are organized by the LBS module and

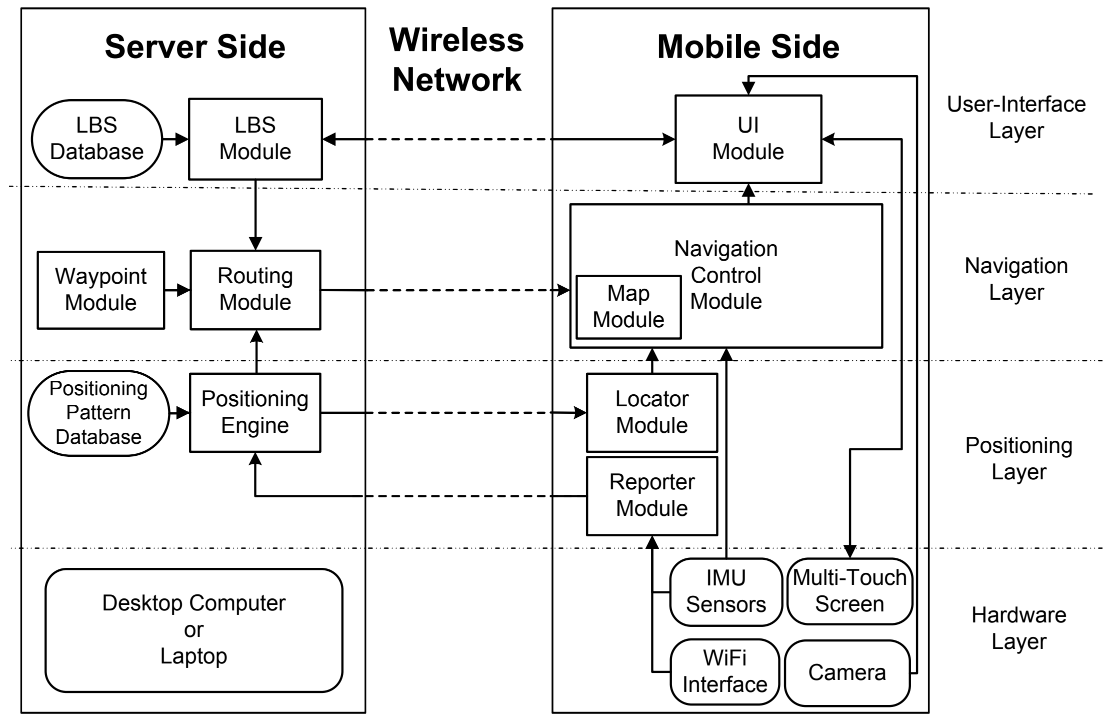


Fig. 4. The proposed system architecture for location tracking, navigation, and mobile AR.

then are sent to the routing module in the navigation layer to transmit to the mobile devices.

B. Mobile Side

The mobile side has two major tasks:

1) Report the RSSs and user's feedback information to the positioning server for positioning and navigation purposes.
2) Display the location-aware service by mobile AR.

To perform the above tasks, the hardware layer requires i) a *WiFi interface* to receive beacons from access points and estimate the RSSs, ii) *IMU sensors* to calculate the user's behavior such as the current moving direction, iii) a *camera* to take pictures from the environment to realize mobile AR, and iv) a *multi-touch screen* for the user to interact with our positioning system.

In the positioning layer, the *report module* will integrate the RSS data and user's feedback information from the WiFi interface and IMU sensors, respectively, and then report the integrated data to the positioning engine in the server side. After the positioning server calculates the user's location, the *locator module* can send the positioning result to the navigation layer. Note that here we use IMU sensors to analyze the user's moving behavior and exploit the analysis data to help correct potential positioning errors due to the signal drifting problem.

The navigation layer contains only the *navigation control module*, which receives the estimated routing path, positioning result, and user's feedback information from the routing module (in the server side), the locator module (in the positioning layer), and IMU sensors (in the hardware layer), respectively. The navigation control module also has a *map module*, which stores the environmental information such as walls and floor plans. Then, the navigation control module rearrange the positioning and navigation results to the user-interface layer.

Similarly, the user-interface layer contains only the *UI (user interface) module*. The UI module will communicate with the LBS module (in the server side) to let the positioning serve know the user's requirement (such as "I want to know where the nearest toilet is?"). Besides, with the pictures of environment from the camera, the UI module can realize mobile AR by adding some auxiliary texts and arrows to give descriptions of the real environment.

IV. Implementation Details

We have developed a prototyping system to verify the practicability of our proposed design in the previous section. In this section, we present our implementation details, including the hardware components of mobile devices, the software architecture of the positioning and navigation system, and the experimental results.

A. Hardware Components

We adopt an HTC magic smart phone [12] as our mobile device. The HTC magic smart phone supports the Android operating system [13] and the GSM (2G), GPRS (2.5G), WCDMA (3G) communications. Besides, it is equipped with an IEEE 802.11 b/g interface to connect with an WiFi network, a GPS receiver to localize in an outdoor environment, a multi-touch screen to let the user interact with the system, a g-sensor to detect the 3D accelerations of the user, and an electronic compass to obtain the user's direction.

The g-sensor is composed of a triaxial accelerometer, a triaxial magnetometer, and a triaxial angular rate gyroscope. The g-sensor can provide the 3D g-values in a range of $\pm 5\,$g, the 3D magnetic field in a range of $\pm 1.2\,$Gauss, and the rate of rotation in a range of 300° per second. The sampling rate of the sensor readings is 350 Hz at most. In addition, the g-sensor can provide its orientation in Euler angle (pitch, roll, yaw) in a rate of at most $100\,$Hz. The optional communication speeds are 19.2, 38.4 and 115.2 kBaud.

B. Software Architecture

The current version of the Android operating system used in the HTC magic smart phone is 1.5 and we use Java to program in the smart phone. On the other hand, we adopt Java to program in the server side, too.

The indoor environment is a multi-storey building deploying with WiFi access points. Each floor of the building is described as a 2D map containing hallways, rooms, walls, stairways and elevators. Then, the positioning server will use such a map as the positioning reference for each floor.

To obtain the location of a user, we write an Android program at the smart phone to periodically collect the WiFi radio signals sent from nearby access points and the measurements reported from the IMU sensors. When a predefined timer expires, a positioning pattern including the RSSs, strides, directions, and user's moving behaviors will be integrated and then this integrated data will be sent to the positioning server through the WiFi network. Then, the positioning server will use this positioning pattern to calculate the location of the user and this calculation result will be sent back to the smart phose to be demonstrated on the screen.

C. Experimental Results

We demonstrate our prototyping system in the 4th, 5th, and 6th floors inside the Engineering Building III of the National Chiao-Tung University, where each floor has deployed with WiFi access points to cover the whole floor. The dimension of each floor is about 74.4 meters by 37.2 meters, which includes walls and rooms. Any two floors are connected with two stairways and two elevators. For each floor, we arbitrarily select 153 training locations along the hallways and inside the rooms. We collect at least 100 RSS patterns at each training locations.

We measure the performance of our positioning system by the positioning errors. For comparison, we also implement one famous pattern-matching localization scheme, the *nearest neighbor in signal space (NNSS)*. During our experiments, each user can receive beacons from averagely 11 access points. In average, the position errors of our system and the NNSS scheme are 3.59 meters and 7.33 meters, respectively, which verifies the effectiveness of our proposed system.

V. Conclusions and Future Work

In this paper, we have proposed using intelligent mobile devices to realize indoor wireless location tracking, navigation, and mobile AR. With the multi-touch screen and the IMU sensors equipped on the mobile devices, we allow users to feedback their moving information to help correct the positioning errors. In addition, with the cameras on the mobile devices, we realize the mobile AR to vividly navigate users in a complex indoor environment. A prototyping system using HTC magic smart phones and g-sensors is developed to verify the practicability of our idea. For the future work, we expect to develop more interesting and content-rich location-based services on our positioning system.

References

[1] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *IEEE INFOCOM*, vol. 3, 2003, pp. 1734–1743.

[2] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a Sentient Computing System," *Computer*, vol. 34, no. 8, pp. 50–56, 2001.

[3] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *ACM Int'l Conf. on Mobile Computing and Networking*, 2001, pp. 166–179.

[4] J. Zhou, K.-K. Chu, and J.-Y. Ng, "Providing location services within a radio cellular network using ellipse propagation model," in *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, vol. 1, march 2005, pp. 559 – 564 vol.1.

[5] P. Bahl and V. N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," in *IEEE INFOCOM*, vol. 2, 2000, pp. 775–784.

[6] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen, "Multidimensional Vector Regression for Accurate and Low-cost Location Estimation in Pervasive Computing," *IEEE Trans. on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1181–1193, 2006.

[7] S.-P. Kuo and Y.-C. Tseng, "A Scrambling Method for Fingerprint Positioning Based on Temporal Diversity and Spatial Dependency," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 678–684, 2008.

[8] T.-C. Tsai, C.-L. Li, and T.-M. Lin, "Reducing Calibration Effort for WLAN Location and Tracking System using Segment Technique," in *IEEE Int'l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 2, 2006, pp. 46–51.

[9] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 869–883, 2008.

[10] L.-W. Yeh, M.-S. Hsu, Y.-F. Lee, and Y.-C. Tseng, "Indoor localization: Automatically constructing today's radio map by irobot and rfids," in *IEEE Sensors Conference*, 2009.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.

[12] HTC Magic. [Online]. Available: http://www.htc.com/www/product/magic/overview.html

[13] Android. [Online]. Available: http://www.android.com/