Research Article

Indoor Location Based VOD Service Using Smart Phone Sensors

Jaegeol Yim and Young-Ho Park

1 Department of Computer Engineering, Dongguk University at Gyeongju, Gyeongbuk 780-714, Republic of Korea
2 Department of Multimedia Science, Sookmyung Women's University, Seoul 140-742, Republic of Korea

Correspondence should be addressed to Young-Ho Park; yhpark@sm.ac.kr

Received 2 January 2014; Accepted 20 February 2014; Published 17 April 2014

Academic Editor: Ruay-Shiung Chang

Copyright © 2014 J. Yim and Y.-H. Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A smart phone is the best device on which a location based VOD service (ILBVS) runs, because a smart phone has PC level computing power and storage capacity, and it is a personal belonging that the owner always carries. Thus, this paper proposes an efficient way of implementing an ILBVS as a smart phone app. ILBVS cannot be realized unless the system can estimate the user’s location. We use a dead-reckoning indoor positioning method that determines user’s location using smart phone sensors.

Another necessary ingredient of ILBVS is a map that is the most important element of the LBS’s user interface. We obtain the floor map from the floor map repository. The most important contribution made by this paper is the method of guessing what the user is most interested in at the current moment. With this method, we implement an example ILBVS.

1. Introduction

Location based service (LBS) prevails throughout everyday life [1]. Navigation, logistics, fleet control, POI (point of interest), and others are all types of LBS. LBS is so useful that it should be available everywhere including inside buildings, subway stations, underground shopping malls, and big city centers [2, 3]. Such LBSs located inside of buildings are called indoor LBS (ILBS) [4, 5]. A WiFi based smart phone with GPS and Gyro sensors provides services to users to support the specific application based program that works in a certain specific zone [6].

LBS, including ILBS, cannot be realized unless the location of the mobile device is identified. Therefore, positioning is one of the essential techniques in LBS development. For outdoor LBS, global positioning system (GPS) provides a general solution for positioning. However, the GPS signal is so weak indoor or underground that we cannot determine a user’s location with GPS alone. Thus, we have to develop indoor positioning methods for ILBS implementation. This paper introduces a dead-reckoning indoor positioning method.

A floor map is the most important element of the user interface of ILBS just like a map is the one of LBS. For LBS, Google map provides a map wherever the user is. However, there is no universal floor map provider. This paper proposes an open floor map repository from which any ILBS system can download any floor map it wants.

This paper proposes a method of guessing what the user is interested in so that our indoor location based VOD service (ILBVS) can play the most appropriate video for the user. The method determines whether the user is standing still or moving around by investigating sensor values. It determines the user’s current location with the dead-reckoning positioning method and the orientation the user is facing. With the location and orientation, it determines what the user is interested in by referring to the floor map.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes
2 International Journal of Distributed Sensor Networks

The deployment of the fingerprinting approach consists of two phases: the off-line phase and the on-line phase. During the off-line phase, they construct a look-up table. An entry of the look-up table is a pair of location identifier and WiFi received signal strength (RSS) at the location. It is known that the fingerprinting approach has a high accuracy [10]. However, there is a major shortcoming of this technique—creating the fingerprint database (look-up table) and maintaining it are not trivial tasks [11].

The range based approach converts the RSS into distance measurements. If three distances between the user receiver and different APs can be obtained, trilateration can be used to estimate the receiver’s position. However, creating an accurate model to convert RSS into the distance is difficult. The propagation of radio signal in indoor environments is complicated. The received RSS from an AP varies significantly (up to 15 dBm) over time at the same location. In addition, indoor environments considerably differ from each other, which means one model may work well for a specific environment but perform poorly in other situations. Hence, it is difficult, if not impossible, to accurately obtain distance measurements from RSSs on a consistent basis [11].

Since smartphones are equipped with a computer and many accurate sensors, developing smartphone applications that count user steps and detect his/her moving direction is possible [12–14]. Anumber of steps can be used to figure out the distance the user has moved. The positioning method that uses the moving distance and direction is called dead-reckoning. A dead-reckoning smart phone positioning approach is free from the shortcomings of fingerprinting and range based approaches.

The authors of [6] introduced their implementation of location based real time reservation service. It automatically recognizes the reservation zone in which the user entered and provides an expected waiting time. Once a reservation is made, it blows an alarm when it is close to the beginning

2. Related Work

Active Badge [7], Active Bat [8], and Cricket [9] are the most representative indoor positioning methods. These positioning methods are accurate enough to be used for practical ILBS system development. However, they require special equipment such as infrared light sensors or ultrasonic wave receivers.

Nowadays, wireless local area network (WLAN) is available almost everywhere. Therefore, the indoor positioning methods based on WiFi signals from the access points (AP) are more economical than those methods that require special equipment. The positioning techniques adapted in WLAN based positioning can be classified into either fingerprinting or range based one.
Wireless sensor networks (WSNs) have several applications in both military and civilian applications. Localization of the sensor node is essential in such networks. Localization is the ability of a sensor node to detect its own position using initial deployment knowledge and the data it gathers from its surroundings. The authors of [15] introduced a beaconless location discovery scheme for a nonflat terrain whereas the authors of [16] proposed a location estimation scheme based on moving nodes that opportunistically exchange known positions. The user couples a linear matrix inequality method with a barycenter computation to estimate its position.

The authors of [17] proposed a novel location technology based shopping service system. It determines user’s location with coordinates of APs and the strengths of signals received from the APs, provides product information, notifies their current location, figures out proper shopping route, and provides the navigation service.

The authors of [18] proposed an algorithm for high precision indoor positioning using lighting LEDs, visible light communication, and image sensors. In the proposed algorithm, four LEDs transmit their three-dimensional coordinate information which is received and demodulated by two image sensors near the unknown position. The unknown position is then calculated from the geometrical relations of the LED images created on the image sensors.
The authors of [19] proposed a moving target tracking algorithm applying the Kalman filter process on the time difference of arrival and the frequency difference of arrival of the measurement signals. Since the estimation error of the Kalman filter process can be large when the a priori noise covariance is assumed with improper values, they offered an adaptive extended Kalman filter to update the noise covariance at every measurement and to find proper noise covariance at each step.

The authors of [13] used the finite state machine (FSM) shown in Figure 1 to count steps. States $S_0$ and $S_1$ represent not walking and possibly started step, $S_2$ and $S_3$ represent if positive and negative peek have been reached, $S_4$ and $S_5$ are used to tolerate the noise, and $S_6$ represents the terminal state. The values they used for the variables are:

- $\text{Thr: 10.0}$, $\text{Pos.\ Peek\ Thr: 11.0}$, $\text{Neg.\ Peek\ Thr: 8.0}$, and $\text{Neg.\ Thr: 9.0}$.

The floor map repository system is an open system to which anybody can upload floor maps and any developer can use a floor map from the system [14]. The floor map repository system provides APIs with which a user can upload the coordinates of any point of interest as shown in Figure 2 [14].

3. Design of the System

One of the main contributions of this paper is to propose an algorithm that guesses what the user is interested in at this moment. If a user is not moving, then we can guess the user is interested in something around the spot where the user is located. It is known that if the standard deviation of recently collected 20 $y$-axis accelerometer values is less than 0.5, then we can conclude that the user is not moving [20].

With the dead-reckoning algorithm introduced in [20], we can determine user’s current location if we know the user’s previous location. However, [20] does not provide the initial location from which we can start applying the algorithm. We propose the following two strategies to initialize the user’s location.
(1) Using the GPS, we keep tracking user’s location. If the GPS value becomes invalid and the user is around a building entrance, then we conclude that the user’s current location is the entrance.

(2) Whenever the user is standing still and there is a POI, we reinitialize the user’s location with the location of the POI.

Once we know user’s current location and the orientation user is facing, we can determine what the user is interested in by referring to the locations of POIs in the floor map [21]. We implement this algorithm as a Web service so that program developers can reuse it.

The workflow of our ILBVS is described in Figure 3. The client begins three activities in parallel: collecting sensor values, rendering the floor map, and playing the video. The rendering activity accesses the floor map repository system to get the map of the domain and then displays the map on an image view.

The video playing activity checks if the user is in “not moving” state. We conclude that the user is not moving if the standard deviation of the recently collected, for 2 seconds, y-axis accelerometer values is less than 0.5.

If the user is determined to be “not moving”, it sends all the collected sensor values to the Web server and invokes the dead-reckoning positioning function. The positioning function counts user’s steps and estimates the user’s location at every step. It finds out which exhibit the user is watching with user’s current location and the direction user is facing. It also finds out the video that is closely related to the exhibit user is watching. Then, the server returns the sequence of estimated locations and the URI of the video. The playing video activity requests a stream of the video on the streaming server we have installed.

We design our indoor location based VOD service (ILVBS) as a client-server system described in Figure 4. In the Web server, the Web services that perform map manipulation and indoor positioning will be implemented. When our indoor positioning Web service determines the
user's location, it refers to the electronic map of the domain. Once it determines the user's current location, it finds the file path of the video that is the most appropriate to the user referring to the video database and sends the file path to the client. Then, the client accesses the Darwin streaming server with the file path to play the video.

The Web services for map manipulation include the following.

(1) registerDrawing() creates a tuple consisting of floor map file name, description, category, and file path and appends the tuple to "drawings" table.

(2) registerDrawingMeasure() creates a tuple consisting of floor map file name, coordinates of the middle point, and scale and appends the tuple to "drawing_measure" table.
cosVal1 = diffOri to radian and to cosine and multiplication step-length

sinVal1 = diffOri to radian and to sine and multiplication step-length

(3) registerAccessPoint() creates a tuple consisting of MAC address, floor map file name, AP’s coordinates, and time of registration and appends the tuple to “accessPoint” table.

(4) deleteDrawing(), given a floor map file name, deletes the tuple designated by the name from “drawings” table.

(5) deleteDrawingMeasure(), given a floor map file name, deletes the tuple designated by the name from “drawing_measure” table.

(6) getAllDrawingName() retrieves all tuples from “drawings” table and returns the results as an ArrayList.

(7) getDrawingDirPath(), given a floor map name, returns the file path of the floor map file.

(8) getScale(), given a floor map name, returns the scale of the floor map designated by the name.

When a new electronic floor map file is uploaded, the server draws a floor image represented by the uploaded file and saves the image file in the database. Our system has “DrawingManager” class in which all the following methods needed to draw floor images are defined.

(1) createAndStoreDrawing() invokes initDrawing(), readLineFromDxfFile(), readFromList(), and drawObject() methods to draw a floor map.

(2) initDrawing() instantiates a BufferedImage object in which a floor image will be drawn and creates a few variables including drawingList, objectIdentifier, xMax, xMin, yMax, and yMin.

(3) readLineFromDxfFile() reads a floor map file line by line if the line contains information of geometric figure and then saves the line in lineList; otherwise it discards.

Figure 10: A part of our program that estimates the current location.
Figure 11: A part of our implementation of the Kalman filter process.

(4) readFromList() retrieves a line from lineList if the line includes "LINE" and then invokes lineModule().

(5) getLineCouple() retrieves two lines from lineList.

(6) lineModule() with LINE information creates a Line object, appends it to drawingList, and appends the shape ID (LINE) to objectIdentifier.

(7) drawObject() removes the first element from objectIdentifier in order to identify the geometric figure type, removes the elements from the head of drawingList to obtain data about the figure including coordinates of the start point and the end point, and draws the geometric figure represented by the data.

(8) windowToPaper() converts window coordinates into map coordinates.

The Web server in Figure 4 provides indoor positioning Web services that determine the user’s current location using the finite state automata described in Figure 1. A sequence of user’s locations represents the trace of the user. A trace
of a user should be smooth. However, results of our indoor positioning are not smooth. Therefore, we apply the Kalman process on our results of indoor positioning.

4. Implementation of an Example ILBS

We have implemented an example indoor location based VOD service (ILVBS), a museum guide. As we have discussed earlier, the server part of this system includes a floor map database and provides floor map file uploading facility. Using MultiParser, our floor map registration program separates the floor map file from accompanying related information such as buffer size, description, and category as shown in Figure 5.

The server draws a floor map image for each floor map file in the map database. By invoking initDrawing(), readLineFromDxfFile(), and so on as shown in Figure 6, createAndStoreDrawing() draws a floor map image and saves it in the database.

Our initDrawing() creates BufferedImage object and initializes variables as shown in Figure 7.

In a DXF file, we can find the following geometric figures: “LINE,” “LWPOLYLINE,” “CIRCLE,” and “ARC.” Our readLineFromDxfFile() collects information of those figures and saves it in an ArrayList called lineList as shown in Figure 8.

Apart of our step counting program is shown in Figure 9. This is our implementation of the finite state automata shown in Figure 1. The variable “cur value”, in the program, is the accelerometer value and the next status is determined by “cur value” and current status.

Figure 10 shows our program that estimates the user’s current location. It calculates the number of pixels the user moved from the previous location per step in 2-dimensional space. The direction that the user is heading is obtained from orientation sensor in Android and saved in diffOri. Depending on the direction, x- and y-coordinate can positively or negatively increase.

We have implemented the Kalman filter process in order to make the user’s trace smooth. A part of our implementation
is shown in Figure 11. It repeats the prediction step and the correction with measurement step.

We have implemented a database for drawings, videos, and users. A part of our database system is shown in Figure 12.

We have implemented a database manipulation class as shown in Table 1. The methods defined in the class are used by our floor map manipulation Web services. Given a name of a floor map, x- and y-coordinates, orientation, and video path, registerMetadata() inserts a new tuple consisting of these elements into drawings_position_table.

Our implementation of the video database manipulation class is shown in Table 2. “allgenre()” retrieves all tuples from video_genre table. “searchVideo (video_name)” retrieves the video_path from “video” table where “video name” of the video_path is equal to video_name. “searchVideo (map, video_no)” retrieves the tuples whose video_drawing_name and video_no are equal to map and video_no, respectively.

The client system of our ILBVS is an Android application. It includes DrawingDeleteActivity, DrawingRegisterActivity, DrawingRegisterAPActivity, DrawingViewerActivity, TraceActivity, TraceKFAActivity, TraceMenuActivity, UserJoinActivity, and UserdeleteActivity.

DrawingRegisterAPActivity has to find the image coordinate of the touched point. Our client renders the image of a floor map on an android: scaleType ImageView and defines the scaleType as MATRIX so that we can use Matrix class in order to move, zoom, and skew the image. Matrix class has many methods of manipulating the image. For example, postScale method of Matrix class zooms the image. Matrix class has 3 x 3 matrix and attributes of the image are recorded in this matrix. Using this information, we can convert screen coordinate (x, y) into image coordinate as follows:

```java
float[] values = new float[9];
matrix.getValues(values);
imageX = x/values[0]−values[2]/values[0];
imageY = y/values[4]−values[5]/values[4];
```

5. Experiments

We have tested our dead-reckoning indoor positioning by walking around the space in the hall shown in Figure 13. The blue dots in the figure represent the marks we have made on the hall. The marks form an oval whose transverse diameter is about 5.8 meters and conjugate diameter is about 2.82 meters. There are 27 marks and the average distance between two adjacent marks is about 0.5 meters. We walked around the space by stepping on the marks while the positioning app was running. Our experiments showed that the average number of steps the app counts per round is about 25.4 and the average...
Table 1: Function description of DrawingDao.

| Function name          | Input                                   | Output          | Description                                                                 |
|------------------------|-----------------------------------------|-----------------|-------------------------------------------------------------------------------|
| registerMetadata       | Map name, object ID, map coordinate of the object, and orientation of map object in the map | Success or failure | It inserts the input into the database                                        |
| getMetadata            | Map name                                | List of metadata | It retrieves metadata of the input map                                       |
| registerDrawing        | Map name, map description, map category, and map path | Void            | It inserts input into "drawings Table"                                       |
| deleteDrawing          | Map name                                | Void            | It deletes the map from "drawings Table"                                    |
| deleteDrawingMeasure   | Map name                                | Void            | It deletes the map from "drawing_measure Table"                             |
| deleteAccessPoint      | Map name                                | Void            | It deletes the map from "accessPoint Table"                                 |
| getAllDrawingName      | Void                                    | List of map names | It retrieves all map names in "drawings Table"                              |
| getDrawingDirPath      | Map name                                | Map path        | It retrieves the file path of the map from "drawings Table"                 |
| registerAccessPoint    | MAC address of the access point (AP), map name, map coordinates of the AP, and date time | Void            | Inserts the input into "accessPoint Table"                                  |
| registerDrawingMeasure | Map name, half of map width, half of map height, and scale of map | Void            | It inserts the input into "drawing_measure Table"                           |
| getScale               | Map name                                | Scale of map    | It retrieves the scale of the map from "drawing_measure Table"              |

error at the end of a round is about 2.34 meters. An example test result is shown in Figure 14.

After testing our indoor positioning, we have tested our ILBVS on the virtual museum shown in Figure 15. There are 11 items (exhibits) in the room. We have performed experiments of visiting exhibits one by one while estimating the user’s location at every step. Estimated user’s locations are represented as red dots in Figure 16. Then, we applied the Kalman filter process on the sequence of locations corresponding to the red dots and showed the result as blue dots in the figure.

We have also tested our location based VOD application. When we walked slowly keeping the step length 0.5 meters and straightly moved from one exhibit to another, our app almost 80% correctly recognized the exhibit the user is watching and played the appropriate video as shown in Figure 17. However, when we were moving around randomly, the app poorly recognized the exhibit the user is watching.

6. Conclusions

The popularity of indoor location based service (ILBS) is increasing rapidly. It will be useful if an ILBS guesses what the user is interested in and plays a video that is closely related to the interested subject. This paper has proposed a system structure of indoor location based VOD service (ILBVS) system that guesses user’s interest and actively plays the most appropriate video.

The correctness of the guessing algorithm is dependent on the correctness of positioning algorithm. Our experimental results showed that the positioning algorithm is erroneous unless the user carefully keeps his/her step length 0.5 meters.

Our test results also showed that the orientation sensor installed in smart phones is also sensitive to environments. When we tested orientation sensors while walking along the walls, the sensor values were significantly wrong. Therefore,
Table 2: Function description of VideoDao.

| Function name | Input                  | Output                                      | Description                                                                 |
|---------------|------------------------|---------------------------------------------|----------------------------------------------------------------------------|
| allgenre      | Void                   | List of all genre IDs and names             | It retrieves all genre IDs and names from "video_genre Table"             |
| searchVideo   | Video name             | Video streaming path                       | It retrieves the streaming path of the video from "video Table"           |
| searchVideo   | Map name and object ID in the map | Video streaming path | It retrieves the video streaming path for the object on the map from "video Table" |
| deleteVideo   | Video name             | Void                                        | It deletes the video from "video Table"                                   |
| registerVideo | Video name, map name, object ID on the map, video pat, and genre ID | Success or failure             | It inserts the input into "video Table"                                   |
| listgenre     | Genre ID and mode      | A genre ID and a genre name                 | If mode is 0, then it retrieves the root genre; if mode is 1, then it retrieves the successor of the input genre; if mode is 2, then it retrieves the predecessor of the input genre from "video_genre Table" |
| Listvideo     | Genre ID               | List of video names and video paths         | It retrieves all the video names and streaming paths of the videos that belong to the input genre from "video Table" |
| searchMetadata| Video streaming path   | Metadata                                     | It retrieves the metadata about the input video streaming path from "video_metadata Table" |

Figure 16: Estimated user’s location.

we are in doubt about the research results of SLAM (simulta- neous localization and mapping) with smart phones.

There are building blue prints for all the buildings. They can open rough building blue prints of public part of the building. We strongly believe that floor maps should be utilized in practical ILBSs until more accurate sensors are installed in smart phones.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2011-0006942) and by “Development of Global Culture and Tourism IPTV Broadcasting Station” Project through the Industrial Infrastructure Program for Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393).

References

[1] P. Chen, J. Gu, X. Lin, and R. Tan, “A probabilistic approach for GNN queries in LBS,” International Journal of Multimedia and Ubiquitous Engineering, vol. 7, no. 2, pp. 189–194, 2013.

[2] Y.-H. Park, K.-Y. Whang, B. S. Lee, and W.-S. Han, “Efficient evaluation of partial match queries for XML documents using information retrieval techniques,” in Proceedings of the 10th international conference on Database Systems for Advanced
[3] S.-Y. Ihm, K.-E. Lee, A. Nasridinov, J.-S. Huh, and Y.-H. Park, “Approximate convex skyline: a partitioned layer-based index for efficient processing top-k queries,” Knowledge Based Systems, 2014.

[4] G. Lee, J. Yim, and C. Han, “Implementation of an indoor-location-based interactive VoD player using web services,” Information B, vol. 16, no. 2, pp. 1309–1316, 2013.

[5] J. Yim, J. Joo, and G. Lee, “Petri net-based ontology analysis method for indoor location-based service system,” International Journal of Advanced Science and Technology, vol. 39, pp. 75–92, 2012.

[6] S. K. Kang, H. K. Kang, J. E. Kim, H. Lee, and J. B. Lee, “A study on the mobile communication network with smart phone for building of location based real time reservation system,” International Journal of Multimedia and Ubiquitous Engineering, vol. 7, no. 2, pp. 17–36, 2012.

[7] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” ACM Transactions on Information Systems, vol. 10, no. 1, pp. 91–102, 1992.

[8] A. Ward, A. Jones, and A. Hopper, “A new location technique for the active office,” IEEE Personal Communications, vol. 4, no. 5, pp. 42–47, 1997.

[9] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket location-support system,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom ’00), pp. 32–43, Boston, Mass, USA, August 2000.

[10] J. Yim, “Introducing a decision tree-based indoor positioning technique,” Expert Systems with Applications, vol. 34, no. 2, pp. 1296–1302, 2008.

[11] L. Chen, B. Li, K. Zhao, C. Rizos, and Z. Zheng, “An improved algorithm to generate Wi-Fi fingerprint database for indoor positioning,” Sensors, vol. 13, no. 8, pp. 11085–11096, 2013.

[12] Y. Chon and H. Cha, “LifeMap: a smartphone-based context provider for location-based services,” IEEE Pervasive Computing, vol. 10, no. 2, pp. 58–67, 2011.

[13] M. Alzantot and M. Youssef, “UPTIME: ubiquitous pedestrian tracking using mobile phones,” in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’12), pp. 3204–3209, Shanghai, China, April 2012.

[14] J. Yim, “A smartphone indoor positioning method,” International Journal of Smart Home, vol. 7, no. 5, pp. 9–18, 2013.

[15] S. Krupadanam and H. Fu, “Beacon-less location detection in wireless sensor networks for non-flat terrains,” International Journal of Software Engineering and Its Applications, vol. 2, no. 3, pp. 55–76, 2008.

[16] G.-D. Kang, T. Pérennou, and M. Diaz, “Barycentric location estimation for wireless network indoors localization,” International Journal of Hybrid Information Technology, vol. 2, no. 2, pp. 1–12, 2009.

[17] J. Liu, T. Chen, J. Bao, J.-U. Kim, and J. Wang, “Location technology-based mobile shopping service system,” International Journal of Hybrid Information Technology, vol. 5, no. 4, pp. 133–142, 2012.

[18] M. S. Rahman and K.-D. Kim, “Indoor location estimation using visible light communication and image sensors,” International Journal of Smart Home, vol. 7, no. 1, pp. 99–114, 2013.

[19] D. Kim, J. Ha, and K. You, “Adaptive extended kalman filter based geolocation using TDOA/FDOA,” International Journal of Control & Automation, vol. 4, no. 2, pp. 49–58, 2011.

[20] J. Yim, “Development of web services for WLAN-based indoor positioning and 8 floor map repositories,” International Journal of Control and Automation, vol. 7, no. 3, pp. 63–74, 2014.

[21] J. Yim, J. Joo, and S. Trimi, “A new method for recognizing objects in photos taken by smart phones,” Journal of Cases on Information Technology, vol. 8, no. 13, pp. 289–299, 2013.
Submit your manuscripts at
http://www.hindawi.com