International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 22 • 2017

Indoor Navigation using BLE Beacons and Augmented Reality

Ruhi Mahajan[a] Rishav Gupta[a] and Shubham Sharma[a]

[a]*Department of CSE, Jaypee University of Information Technology Solan , India*

E-mail: mahajan.ruhi214@gmail.com, rishavgupta2308@gmail.com, sharmashubham2341@gmail.com

Abstract : The main idea of our proposed work is to give an implementation of indoor navigation using augmented reality and BLE (Bluetooth low energy) beacons. Augmented Reality is a promising technology in which one's opinion of the real-time environment is improved by superimposing information generated by computer like textual data, audio as well as graphical data and objects onto a display screen. The proposed work is basically a mobile based android application which will be compatible with all the android versions having version API level above 16 (Jellybean).

In particular, this android application explicitly addresses issues caused by the relatively inaccurate GPS and orientation sensors during navigation by using beacons which interact with the android application using an API.

Keywords : *Augmented Reality, BLE Beacons, Android.*

1. INTRODUCTION

Augmented reality is a live view of physical and real environment whose elements are supplemented by sensory input like video, sound, GPS data or graphics generated by computer. Augmented Reality converts the environment around us into a digital interface with the help of introduction of virtual things in real time in the physical and real world Augmented Reality is basically of two types: Marker less and Marker based. The marker-less based augmented reality uses the positional data such as the GPS, compass or the accelerometer, and then performs the operation. The marker based augmented reality uses the camera, scans the visual cues, and after finding the specific cue it performs the action.

While today's more sensitive GPS chips present in the recent mobile phones can sometimes navigate a user inside a building by receiving signals from enough satellites, but typically the position is not exact enough to be useful. The signals received from the satellites are weakened and scattered by the walls, roofs as well as by the other objects present inside the building. Besides, the inaccuracy measured by many GPS chips can be larger than the indoor space itself. The GPS chips present in the recent mobile phones can give an offset of about 40 meters In particular; the proposed work explicitly addresses issues caused by the relatively inaccurate GPS and orientation sensors during navigation by using beacons which interact with the android application using an API. Our work highlights the use of beacons to retrieve the user's current location. The major advantage of

using beacon is that the user will have a seamless experience; the user is not required to scan any marker for augmented reality to work. Hence the efficiency is higher and also the time taken to respond to user query is less than other mechanisms. Bluetooth beacons are basically low energy hardware transmitters and their main aim is to broadcast their identifier to close by portable devices. The main function of beacons is to transmit a unique identifier value that will be chosen by an operating system or an appropriate application with the help of Bluetooth energy proximity sensing method. Beacon is a 1-way transmitter to the receiving device or smart phone.

2. PROPOSED WORK

In our research paper we are going to implement an Indoor navigation system with the help of Bluetooth low energy beacons and augmented reality concepts. Firstly we will make the beacons which will emit the low-Bluetooth energy signal which will be detected by the API in the android application. The next step is creating the UI design of the android application. We create an activity in the android application in which the user will have to feed the destination. The entered destination should match with the destinations in the database, and then the current location of the user will be detected by capturing the signals from the nearest beacon which in result will navigate the user using augmented reality. The next step which comes is the Indexing of the beacons present in the database. This module will insert the MAC of the beacon which is at the destination, and the MAC of all the beacons which the user will encounter during the navigation. Then depending on the current location of the user, the shortest path to reach the destination will be calculated.

2.1. Making Beacons

The first step of our proposed system consists of making a beacon. The components required to build the beacon consist of Atmel Atmega 328P/8 IC, nrf24l01+ module and a coin battery. We will use the Arduino Uno as an ISP (In-system programmer) to upload the code on the Atmel Atmega 328P/8 IC. After programming the IC, we will connect it with the nrf module and a battery to obtain the beacon.

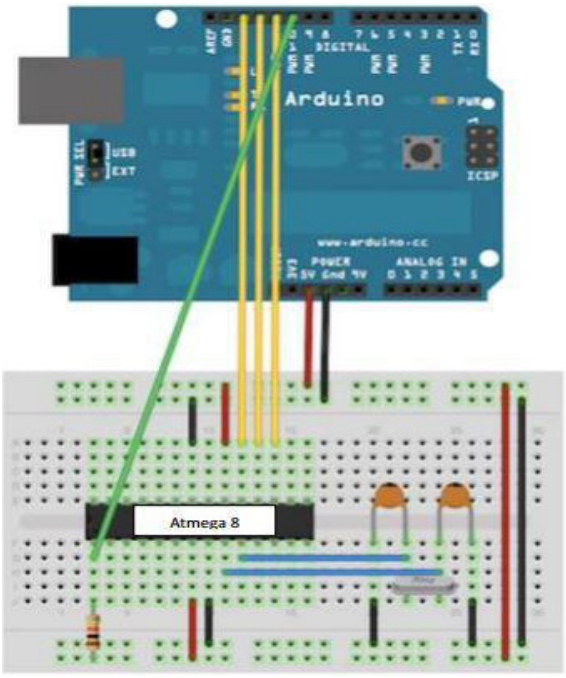

Figure 1: Using Arduino as an ISP

2.2. API Design

The system design consists of beacon, API (to discover the beacons) and an android application. The process of making the beacons is already explained. The API (Application programming interface) consists of two classes namely Beacons.java and OctiseBeacon.java.

The Beacons.java class contains functions like :

1. getRssi(): to return the Rssi(received signal strength intensity)
2. getMac(): to return the unique MAC of the beacon discovered
3. getName(): to return the name of the beacon
4. getMiss(): to return a Boolean variable which tells about whether it was hit or a miss in detecting the beacon.

   Other functions include setRssi(), setMac(), setName(), and setMiss().

The OctiseBeacon.java class contains the following functions:

1. CheckBLEcompatibility (): Checks whether the android version is compatible with BLE (Bluetooth low energy) devices or not. The return type of the function is Boolean; return true when compatible otherwise false.
2. CheckBluetoothcompatibilty (): Checks whether the android version is compatible with Bluetooth or not. The return type of the function is Boolean; return true when compatible otherwise false.
3. BluetoothEnabled (Activity activity): Checks whether the Bluetooth adapter is enabled or not. The function prompts a message to the user if Bluetooth adapter is not enabled. The return type of the function is void.
4. ScanLeDevice (final Boolean enable): The function calls the call back which scans and detects the BLE beacons. The return type of the function is void.

The next thing which comes is the handling of the scanned data. The received rssi values of the beacons are not smooth, hence it requires smoothening. This is achieved by using the getProcessedBeaconData(Beacon[] beacon)function.

This function maintains 3 dictionaries namely: nameDic, newDevs and miss. Whenever the searched data is returned to this function, all the beacons are added to the nameDic dictionary which is uniquely identified by the MAC. After the first search, the entire next search's data is added to the newDevs dictionary which is also uniquely identified by the MAC. If the newDevs dictionary contains a beacon which is not in the nameDic dictionary, it is added to the nameDic dictionary. If the newDevs dictionary contains the beacon which is in the nameDic dictionary, then the value with the lower rssi is considered for accuracy. All the devices which were previously found but were not found in the next search are added to the miss dictionary which is also uniquely identified by the MAC. If there are consecutive three misses, then the device is removed from the list. After this processing, rolling average is calculated in order to refine the result and then the processed data is returned.

$$\text{rolling\_avg} = ((\text{count} * \text{dic.get(key)}) + \text{newDevs.get(key)}) / (\text{count} + 1); \quad (1)$$

The key used is the MAC, which is unique to every beacon. The variable count basically tells us about the number of searches performed in a particular time span.

2.3. System Design

The system design is very simple and consists of three main components:

1. Beacon which will emit the low Bluetooth energy signal
2. An API (Application Programming Interface) whose main purpose is to detect the beacons and return the processed data
3. An android application which will help the user to navigate indoor.

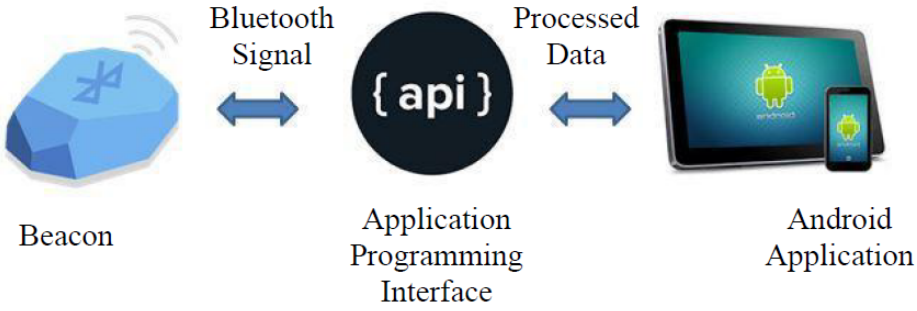

Figure 2: System Design

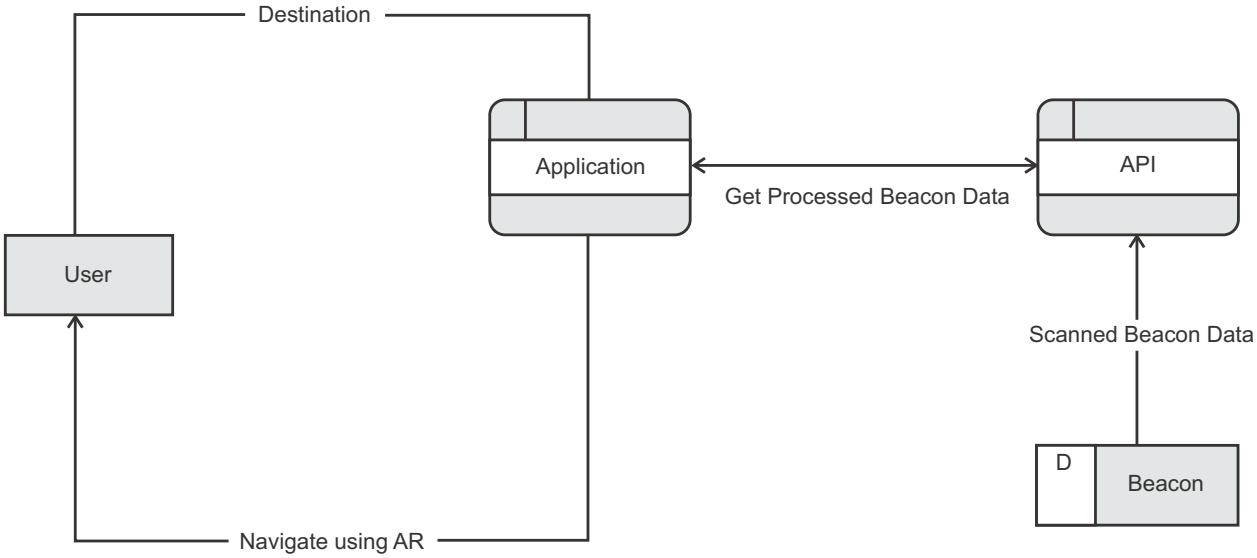

Figure 3: Data Flow Diagram

First of all, the user will be required to fill in the destination. Then we will scan for the nearest beacon and then calculate the minimum distance using the known data and by using algorithms such as Prim's or Dijkstra's algorithm. After calculating the minimum distance, we will get the user direction using the compass of the phone. By getting all the above parameters we can navigate the user. In case the user goes in wrong direction, then the nearest beacon will be scanned and from that beacon shortest distance to the destination will be calculated & the user will be navigated. Since, we are using beacons here, the user will have a seamless experience because the scanning will be done in the background and there is no requirement of scanning the marker as in case of marker based augmented reality. Hence greater efficiency can be achieved. The architecture of the proposed system is shown below:

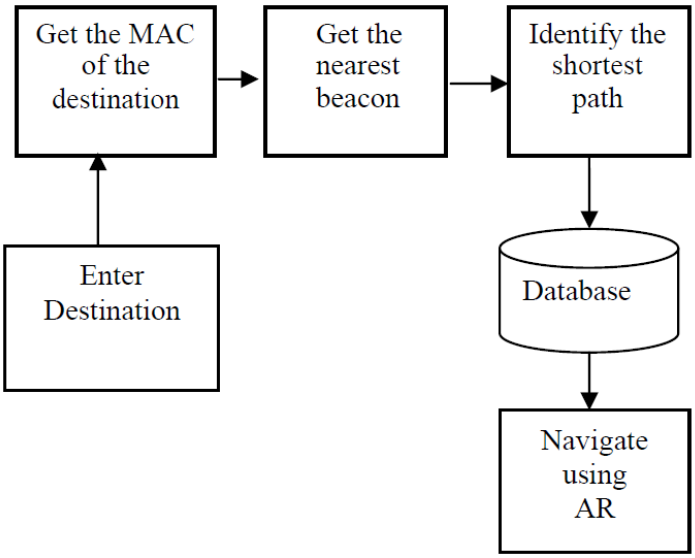

Figure 4: Architecture of Proposed System

3. RESULTS AND ANALYSIS

We have tested our system within our campus with different values of distance and the initial results yielded accuracy of about 63.1%, but after calibrating the beacon the accuracy was about 85.7%. The Table 1 & Figure 5 given below shows the results for different distance values and RSSI values.

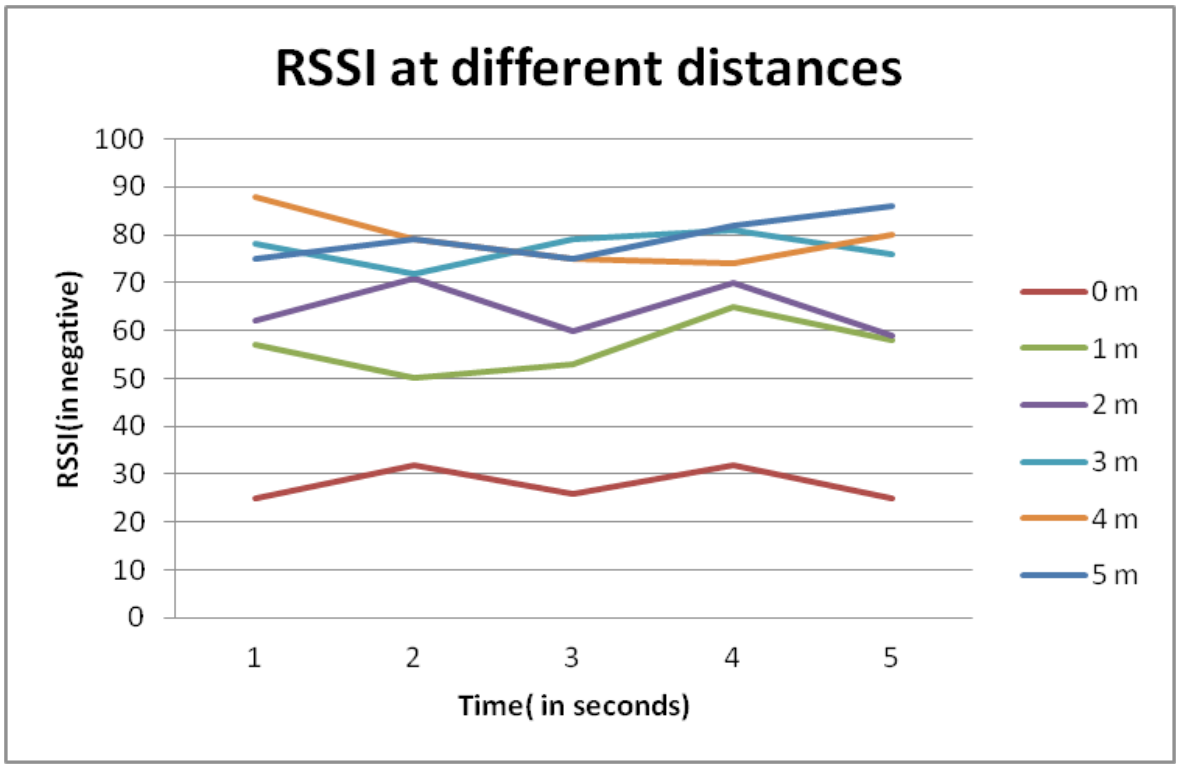

Figure 5: Received RSSI at different distances

Table 1
Experimental Results

Test ID	*Distance*	*Result*
1.	0.5 m	Beacon is in the range of API
2.	1m	Beacon is in the range of API
3.	1.5 m	Beacon is in the range of API
4.	2 m	Beacon is in the range of API
5.	3 m	Beacon is in the range of API
6.	4 m	Beacon is in the range of API
7.	5 m	Beacon is in the range of API

4. CONCLUSION

Indoor navigation deals with navigation within buildings. Because GPS reception is normally non-existent inside buildings, other positioning technologies are used here when automatic positioning is desired. WiFi or beacons (Bluetooth Low Energy, BLE) are often used in this case to create a so-called "indoor GPS". BLE beacon is useful in various aspects. The project highlights the use of beacons to retrieve the user's current location. The major advantage of using beacon is that the user will have a seamless experience; the user is not required to scan any marker for augmented reality to work. Hence the efficiency is higher and hence the time taken to respond to user query is less than other mechanisms. We were successfully able to design the BLE beacon, which will transmit the Bluetooth low energy signals which can be detected by any android device running on Android version greater than 4.0(Ice cream sandwich). We were also successfully able to design the API (Application program interface) which will collect all the scanned beacon data, process the data and then return the processed data to the application.

5. FUTURE SCOPE

In our proposed work we are using beacons for only detecting the user's location, but beacon can be used in other applications too. Some applications of beacons which can be implemented in future in restaurants and homes are mentioned below:

1. Beacons can be used to restrict device access. Whether the user is in the range of the BLE beacon, the device will remain open.
2. Beacon can be used as an assistant for disabled people.
3. Beacon can be used for seamless control for switching on/off electrical devices.
4. Beacon can be used for security purposes and can help in preventing hazards.
5. Beacon can be used to trigger notification for discounts on food, special menu, etc in a restaurant.
6. Beacon can be used for surveying, real time feedback and contactless payment in a restaurant.
7. Beacon can be used by the customer to order ahead.
8. Beacons placed at particular locations can be issued a particular app, and the notification can be triggered to the user and user can open it directly.

REFERENCES

[1] Patkar, Raviraj S., S. Pratap Singh, and Swati V. Birje. "Marker based augmented reality using Android os." International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) 3.5 (2013): 64-69.

[2] Renukdas, Prasad, Rohit Ghundiyal, H. Gadhil, and Vishal Pathare "Markerless augmented reality android app for interior decoration." International Journal of Engineering 2, no. 4 (2013).

[3] Tokusho, Yoshitaka, and Steven Feiner. "Prototyping an outdoor mobile augmented reality street view application." Proceedings of ISMAR Workshop on Outdoor Mixed and Augmented Reality. Vol. 2. 2009.

[4] Geiger, Philip, Marc Schickler, Rüdiger Pryss, Johannes Schobel, and Manfred Reichert. "Location-based mobile augmented reality applications: Challenges, examples, lessons learned." (2014): 383-394.

[5] Jung, Joonyoung, Dongoh Kang, and Changseok Bae. "Distance estimation of smart device using bluetooth." Personal Computing Platform Research Team (2013): 13-18.

[6] Reitmayr, Gerhard, and Dieter Schmalstieg. "Location based applications for mobile augmented reality." Proceedings of the Fourth Australasian user interface conference on User interfaces 2003-Volume 18. Australian Computer Society, Inc., 2003.

[7] Sun, R., Y. Sui, R. Li, and F. Shao. "The Design of a New Marker in Augmented Reality. IPEDR vol. 4." (2011).

[8] Feldmann, Silke, Kyandoghere Kyamakya, Ana Zapater, and Zighuo Lue. "An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation." In International Conference on Wireless Networks, pp. 109-113. 2003.

[9] A. Awad, T. Frunzke, and F. Dressler, "Adaptive Distance Estimation and Localization in WSN using RSSI Measures," 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, Aug. 2007, pp. 471-478.

[10] Dnyanoba, Birajdar Amar, B. Nagajayanthi, and Prakash Ramachandran. "Development of an Embedded System to Track the Movement of Bluetooth Devices based on RSSI." Indian Journal of Science and Technology 8.19 (2015).

[11] Khan, Dawar, Sehat Ullah, and Ihsan Rabbi. "Factors affecting the design and tracking of ARToolKit markers." Computer Standards & Interfaces 41 (2015): 56-66.

[12] Hightower, Jeffrey, et al. "Design and calibration of the spoton ad-hoc location sensing system." unpublished, August 31 (2001)