

Real-time Indoor Map Generation to Identify Spatial Environment Changes

J. A. D. C. Anuradha
Jayakody

Department of Information
Systems Engineering, Sri Lanka
Institute of Information
Technology,
Sri Lanka

R. H. Suriarachchi
Department of Software

Engineering
Sri Lanka Institute of Information
Technology
Sri Lanka

O. W. Samarasinghe
Department of Software

Engineering
Sri Lanka Institute of Information
Technology
Sri Lanka

B. A. Chamain

Department of Software Engineering
Sri Lanka Institute of Information Technology
Sri Lanka

A. H. M. C. M. Abeyrathna

Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

ABSTRACT

While there are many well-established techniques for outdoor contexts using technologies such as GPS, GSM and GNSS, they cannot be applied to an indoor environment. Thus, Indoor navigation remains a challenge as the state of an indoor environment is always unpredictable. Geographical co-ordinates can be applied for indoor localization, but they are not useful inside a building with multiple floors. In most of the approaches, a blueprint of the building is needed for mapping purposes. Hence, there is a need for an improved approach for indoor localization and mapping to overcome the drawbacks stated above. This paper presents an approach to generate a real-time indoor map in-order to identify the changes in the spatial environment. In this approach, authors consider an approach where the information is gathered from users' mobile devices. The position of the user is obtained using iBeacons. Furthermore, this paper discusses how to obtain the sensor data and position of the user by using iBeacons, transferring the data to the database via an API, detecting objects present in the indoor environment and generating the real-time map.

Keywords

Components; indoor navigation, localization, iBeacons, API, SLAM

1. INTRODUCTION

With the development of complex buildings such as shopping malls, libraries, there is a need for a proper way of navigating inside those buildings. Therefore, real-time mapping for the indoor spatial environment is one of the most important research areas today. An indoor environment is ephemeral as the positions of the objects can be changed within a short period. People in the indoor environment are required to be informed about such changes when they are navigating. It is difficult to predict the position of the user as the geographical co-ordinates are not accurate in an indoor environment. Many researchers have used Wireless local area network (WLAN) which has a wide range of utilization in moving from one location to another location and in the newest technology, Bluetooth (iBeacons) [1, 2]. Unlike the Global Positioning System (GPS), iBeacons has the capability of tracking micro locations even in a limited geographical area in a very accurate manner. Since it works at a long distance, this makes it more beneficial.

Some researchers have come up with solutions that require structure/blueprint of the building. Some of those approaches will not consider the dynamic changes in an indoor environment. For instance, present buildings are usually partitioned using partition walls. Those partitions can be changed at any time. Those changes should be updated in the map as soon as it is completed. Therefore, researchers have come up with a different solution that uses highly expensive hardware equipment such as motion detectors, laser scanners and RGB-D cameras [3]. This type of hardware should be installed or mounted on a robot.

Authors propose a novel solution to overcome this problem by using Android smartphones sensors. The proposed approach is an android based mobile application "AssistMe" which generates the real-time map for the particular indoor spatial environment using crowd-sourced data. Information that is needed to generate the map will be gathered from users' mobile devices and sent to the cloud server through API. Data storing, processing and generating a map are done on the cloud server. The generated map will be sent to the mobile device via provided API.

2. RELATED WORKS

Simultaneous Localization and Mapping (SLAM) is a technique which simultaneously constructs a map of an unknown environment while navigating. It is also known as Concurrent Mapping and Localization (CML). SLAM is used by autonomous vehicles to navigate in an unknown environment. In the beginning, the position of the map or the vehicle is unknown. When the vehicle starts moving in an unidentified environment, the kinematic model which is present in the vehicle will be populated with landmarks which are present in the environment [4]. A concurrent estimation of the vehicle and the landmarks are needed in this scenario. To implement this, a mobile robot and a range measurement device are needed, but they are expensive and challenging to operate.

There are many types of researches done concerning the topic of indoor localization and navigation within the past years. Authors Jianho Du, Yongsheng Ou and Weihua Sheng have proposed a solution which creates 3D maps of the environment using RGB-D sensor data and motion sensor data. It is a low-cost solution which addresses the issues caused by expensive hardware equipment. In this approach, they have used a multi-level ICP algorithm which eliminates the failure caused by the frame-by-frame alignment if the movement between the two frames is too

large. Nevertheless, there are some issues they have come across in this approach. A large memory is needed to store the data for processing, thus limiting the size of the map generated [5].

There are many indoor navigation techniques which have been done using RGB Depth cameras. The typical failure in this approach is the insufficient overlapping between the frames. Authors Chenglu Wen, Ling Qin, Qingyuan Zhu, Cheng Wang, and Jonathan Li have developed a system to overcome this drawback. In this system, they have integrated a 2-D Laser scanner and an RGB-Depth camera. A fusion-iterative closest point algorithm is used to combine the 2-D mobile platform pose from a Rao-Blackwellized particle filter estimation. An ICP and a generalized-ICP have been proposed by them to achieve effective frame alignment here [6].

Artur Janowski and Michał Bednarczyk have researched on a low cost indoor navigation approach. Their approach is similar to localization done using Wi-Fi signals. They have used Bluetooth devices for the indoor positioning. Their approach has increased the accuracy of the location compared to the traditional trilateral approach. The reason for this is that they have iterative validating algorithms to address the known issues of the signal propagation and its power measurement [7].

You Li, Yuan Zhuang, Haiyu Lan, Peng Zhang, Xiaoji Niu, and Naser El-Sheimy have introduced a method for indoor navigation which uses smartphone sensors and magnetic features. They have integrated Dead-reckoning based on smartphone sensors and magnetic matching based on indoor magnetic features. Several approaches have been used to enhance the magnetic matching including dynamic time warping, weighted k-nearest neighbor, and magnetic gradient fingerprints [8].

J.A.D.C. A. Jayakody et al. have proposed a novel schema design to assist vision impaired in an indoor navigation environment. This approach gets the input through image processing. The labels in the images of the spatial environment can be identified by using a set of pre-defined rules. The images are collected through the API, and then they are sent to the database which has five tables. Object_info table is used to keep the details of the object such as description, category, and received time. Label_info table is used to keep the information about specific label information the user visits. Movement_info table is used to keep the data received from mobile phone sensors. It contains the direction of the object, and angle of the object. Direction_info table contains the path information of each object relative to the given user. The localization_info table contains data received from mobile phone sensors such as coordinate data [9].

Many authors have presented many solutions relevant to the indoor navigation, but most of them have addressed only one aspect such as localization, mapping or indoor positioning. In the approach, which is suggested by authors, are attempting to cover all of the challenging aspects as discussed in the problem identification section. A real-time map will be generated from the data obtained using image processing also proposed,

3. PROBLEM IDENTIFICATION

There are many complex building structures available today such as libraries, shopping malls and supermarkets; there is a need for a proper way of navigating in these building. Therefore, real time mapping for the indoor spatial environment is one of an essential areas today. Although there are many location-based services for outdoor environments using geometrical co-ordinates available, they cannot be applied successfully to the indoor spatial environment. Geometrical co-ordinates are not useful in an indoor environment when it comes to buildings with multiple floors. An indoor environment is always unpredictable as objects can change their position from time to time. Therefore, the positions of the objects are needed to be updated real time. Many researchers have come up with different solutions, but they are very complicated and expensive.

With the evolution of technology, usage of android smart phones is rapidly increasing. Therefore, rather than using expensive hardware equipment to collect information of the spatial environment, information can be gathered from sensors and cameras present in smartphones. There is a need for an approach that obtains information by using the sensors available in smartphones and provide instructions for indoor navigation. As only the mobile devices are considered, there are some drawbacks due to poor processing power and battery consumption. Therefore, authors have considered a hybrid mobile application which has its processing and database located in a cloud server.

4. SYSTEM OVERVIEW

The end-user of the “AssistMe” mobile application can be either a building owner or a person who is navigating inside the building. When the user opens the application, processes for IBeacon ranging, data uploading, obtaining device direction and updating the map starts. IBeacon ranging is used to obtain the position of the user. Sensors available in mobile devices such as accelerometer and magnetometer are used to obtain the direction the user is situated with respect to the north. Images of the indoor environment is obtained using mobile device camera. This gathered information will be temporarily stored in the device storage until they are sent to the server to process. When sending the gathered data to the server the API will filter redundant data. The API provides an interface for user authentication and requesting the generated map. Once the data is stored in the server, they will be processed so as to obtain the information required to generate the map. The images obtained will be processed in-order to identify the objects present in the indoor environment. The position of the identified objects will be calculated relative to the direction of the device. The extracted data will be stored in the database along with the positions of the objects. The map will be generated according to the obtained data and the generated map will be sent to the user via API. Fig 1 depicts the system overview.

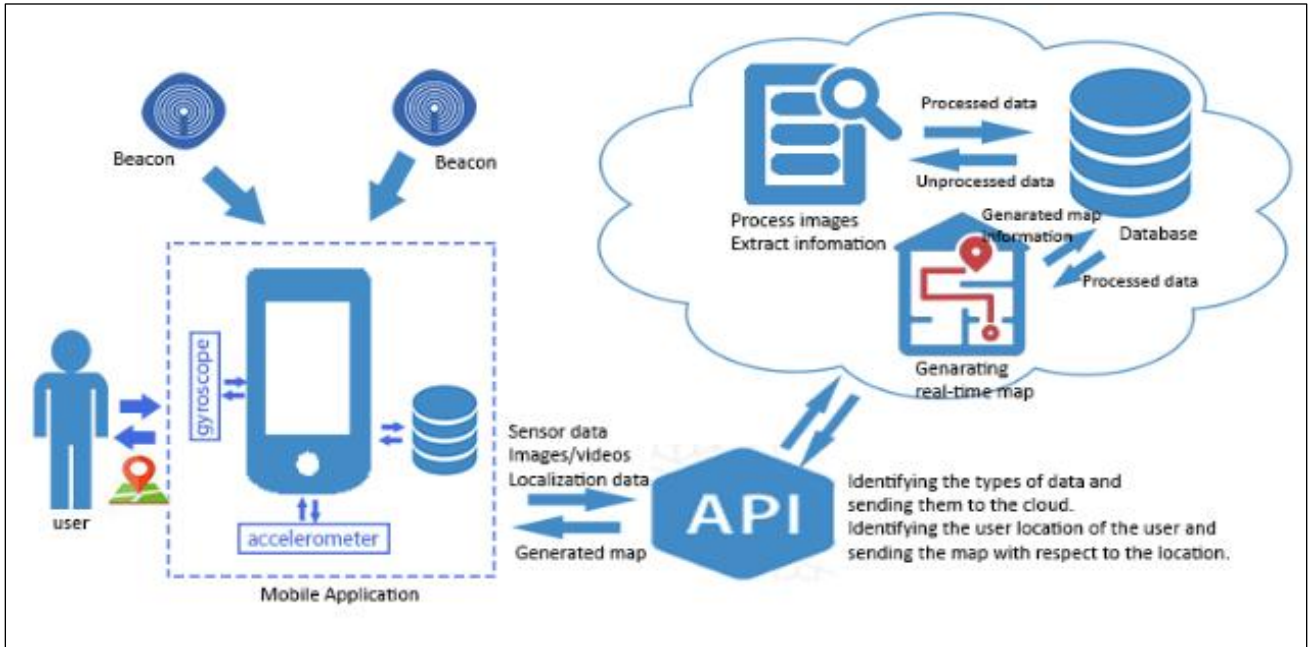


Fig 1: System Overview

4.1 Mobile Application and Localization

Since GPS is not suitable in the Indoor positioning system BLE (Bluetooth Low Energy) technology is used in this approach. Estimote iBeacons are used (Firmware Version: 4.9.4, Hardware Version: F3.3t) for localization. IBeacon is a small gadget which can be attached to a wall, and it emits a signal from time to time. This packet contains the UUID, major, minor, transmitter power, RSSI value, light intensity and temperature [10]. The major value is used to identify a specific location such as an area or a floor in a building. The distance between the device and the IBeacon can be calculated using RSSI value and TX power. Formulas mentioned below are used to calculate the distance between the device and the object [11].

$$RSSI = -10\log d + A \quad (1)$$

Here d is equal to distance and A is equal to TX power.

$$d = 10^{(A-RSSI)/10n} \quad (2)$$

Therefore, the Trilateration Algorithm is applied for obtaining the user position as X and Y coordinates. This position is obtained using three or more IBeacons and the position of each IBeacon.

4.1.1 Trilateration

When there are three or more IBeacons with their positions (x_1, y_1) , (x_2, y_2) and the distance between the device and each IBeacon (d) the device position can be obtained using the Trilateration algorithm.

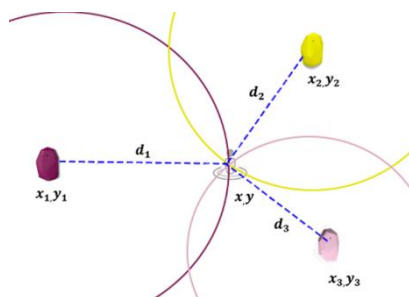


Fig 2: Trilateration

$$(x_i - x)^2 + (y_i - y)^2 = d_i^2 \quad (3)$$

Three equations can be generated using the above equation for each IBeacon. By solving those obtained equations, x and y values can be obtained. This is the position of the user.

4.1.2 Direction

The direction the user is facing is required to obtain the positions of the objects. The data obtained by the accelerometer and the magnetometer is used for this. The accelerometer is used to identify the gravity because the direction of the magnetic field, as well as the horizontal plane, is needed. The unnecessary z -coordinate will be filtered. Therefore, device orientation will not affect when finding the direction. The magnetometer is used to obtain the pitch and roll [12]. Here android SensorManager class provides getRotationMatrix and getOrientation methods to get the direction relative to the North.

4.1.3 Capturing Images

Images are captured using the device camera [13]. This is a background process. From time to time device will capture images and send them to the sever using provided API along with the position details, direction, timestamp, location details and user id. The image will be converted to the base64 format before sending it to the server.

4.1.4 User Login

User authentication is required to track each user and to ensure the privacy of each user. When the user logs in to the system, a session will be created, and the user ID will be sent to the server along with the other data.

4.2 API

In “AssistMe” application communication between mobile device and the server happens through the application programming interface (API). API is built using Java jersey open source framework. “AssistMe” collects the raw data using mobile device sensors and camera. The collected data is sent through the API to the server. This raw data can contain redundant data (duplicated images). API can filter that duplicate data using image pixel matching, image hash matching and image feature matching algorithms. The image blur detection algorithm is used for detecting blurred images.

The general idea of the image pixel matching algorithm is very simple as it compares pixel by pixel of the images. The algorithm gets the color of every pixel of the two images and compares the pixels that have the same coordinates within the image. If the color of each pixel of both images are the same, it considers the two images to be identical. Image hash matching algorithm works like an image pixel matching algorithm. First, simplify the image by reducing its size and colors and Get the average pixel value (the percent gray). Then generate the hash by comparing each pixel's value to the average. After that, it compares the hash of two images to identify if both images are similar. In the feature matching algorithm, it detects certain feature patterns, matches the patterns to detect the similarity of both images.

The image blur detection algorithm uses the *variance of the Laplacian method*. First, it simply takes a single channel of an image (presumably grayscale) and convolve it with a 3 x 3 kernel and then take the variance of the response. If the variance falls below a pre-defined threshold, then the image is considered blurry; otherwise, the image is clear.

4.3 Object Identification

The data which is gathered from the mobile devices need to be processed so that the objects present in the indoor environment can be obtained. The images obtained from mobile device camera is processed and objects which are present in the indoor environment are identified. These objects are identified using tensorflow. Tensorflow is an open source machine learning library, and it starts with an existing classifier called inception. Therefore, there is no need to train the classifier from scratch. Once the object is identified its position is calculated using the direction the device is facing and the distance from the object to the device. Authors use trigonometric equations for this.

Equations (4), (5), (6) and (7) are used to calculate the position of the objects. Θ is the angle of the mobile device in degrees relative to north. d is the distance from the object to the camera. P is the position of the object.

When $0 < \Theta \leq 90^\circ$;

$$P = (x + d \sin \Theta, y - d \cos \Theta) \quad (4)$$

When $90^\circ < \Theta \leq 180^\circ$;

$$P = (x + d \sin (180 - \Theta), y + d \cos (180 - \Theta)) \quad (5)$$

When $180^\circ < \Theta \leq 270^\circ$;

$$P = (x - d \sin (270^\circ - \Theta), y + d \cos (270^\circ - \Theta)) \quad (6)$$

When $270^\circ < \Theta \leq 360^\circ$

$$P = (x - d \sin (360^\circ - \Theta), y - d \cos (360^\circ - \Theta)) \quad (7)$$

4.4 Database

As an enormous amount of data is collected in the process and a NoSQL database is used to store these data. The database is implemented using MongoDB as it supports JSON specifications and has a good scalability [14].

Authors have used the following schema for storing the data obtained from mobile devices.

4.4.1 Mobile_Info

Table 1. Mobile_Info

ID:String	Location:String	Image:String	Position		Direction: Integer	TimeStamp: String
			X: Float	Y: Float		

This table contains raw data obtained by the mobile devices. It consists of image: the path of an image obtained from the mobile device, the position of the user, direction: the angle of the device relative to north and the time stamp: the exact time server received the data.

4.4.2 Object_Info

Table 2. Object_Info

ID:String	Location:String	Name:String	Position		Direction: Integer	TimeStamp: String
			X: Float	Y: Float		

This table contains the details of the objects identified. It consists of the time stamp, location, the name of the object, position and the direction the object is facing.

4.4.3 Beacons_Info

Table 3. Beacons_Info

ID:String	Location:String	Position	
		X: Float	Y: Float

This table contains the positions users obtained using iBeacons. This table is used to identify the paths in the map concerning each location.

4.4.4 Map_Info

Table 4. Map_Info

ID:String	TimeStamp: String	Location:String	Map_Image: Binary
-----------	-------------------	-----------------	-------------------

This table contains the generated map of the indoor environment. It consists of the time_stamp; the exact time the map was generated; location of the map generated and the map_image and the binary object of the generated map image.

4.5 Generate Real-Time Indoor Map

Once the required data is extracted from the image processing is updated in the database, it will similarly be updated in the map. The authors have decided to use Java 2D graphics to develop the map using these data. There are separate classes to draw predefined objects such as doors, tables, chairs and window. Unidentified objects will be shown as obstacles.

As soon as the database is updated the mapping tool has to identify whether the updated records are about a location which was already existed in the database. If the location is already there in the database, the information received will be analysed first. They can be the details of the update of an object, new obstacle information or information totally different from the already existing data. If the information received is entirely different from the prevailing information, it will be ignored. If the location details are unavailable in the database, the new information will be inserted at that particular location. The map

will be generated relative to the x, y co-ordinates already available in the database and saved it into image format.

In addition to that it also detects the user's path, identifies it as a walking path and those details will be drawn on the map. The map will be labelled according to the image processing data obtained after analysing the data gathered from smartphones. For the labelling, the matrix layer will map on top of the image by considering pixels of the image. Then the specific pixel can be identified by value. This is very important in indoor navigation because the user's position can be indicated on the map by using matrix values. The records of the generated map will be stored in a separate table. According to this process, the map gets developed gradually from the crowd source data. Finally, the created map will be displayed on the user's mobile device through API.

Fig 3 depicts the paths identified using iBeacon data. It can be used for navigation purposes. Fig 4 depicts the map generated after processing the extracted data

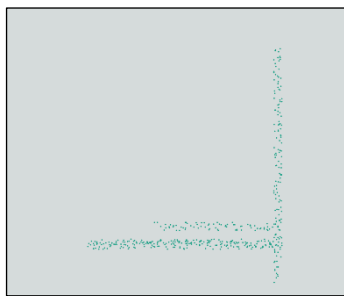


Fig 3: Path generated using iBeacons data

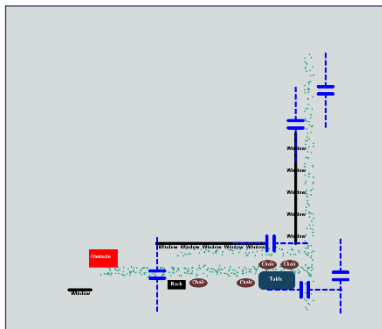


Fig 4: Generated Map

5. TESTING, EVALUATION AND DISCUSSION

RSSI value is unstable due to the influence of radio waves such as absorption, interference and diffraction. Therefore, it is impossible to obtain correct distance using RSSI value. Accuracy is insufficient.

To overcome this problem, an error correction algorithm was used. Therefore, a method was implemented to calculate the distance by considering a variation of RSSI value concerning the distance.

20 sample data were collected with 1 to 10 meters of distance when TX power equals to (strong) 10db. Then the mean was calculated.

Table 5. Median RSSI Table

Distance (m)	RSSI
1	-63
2	-75

3	-88
4	-78
5	-81
6	-79
7	-89
8	-88
9	-91
10	-92

Non-linear least squares technique is used to fit those data into a power curve as in (8) [15].

$$d = A x (r/t)^B + C \quad (8)$$

Here r equals to RSSI value and t is the TX Power. In order to find the A, B and C coefficient the regression analysis was performed and papered the equation.

$$d = 0.726224258 x (r/t)^{5.574242115} - 0.04682180135 \quad (9)$$

Using (9), distance from the iBeacon to the mobile device can be obtained.

Testing API is different from testing other software as no GUI is available for the API. After configuring the server and the database, the postman was used to send API requests to observe the output. The drawback of "AssistMe" API is the less accuracy of the image comparison algorithm.

When detecting objects sometimes false positives were given. Objects which were not present in the environment were detected. The reason for this was that the dataset used for training object detection was not sufficient. Once the dataset was increased authors were able to get more accurate results.

When generating the map, there can be the same object with different co-ordinates obtained by different users. Therefore, the same object will be drawn in different positions. To prevent this, authors collected all records within the same range of location after identifying the object. Then the position of that object was predicted using a test prediction method.

6. CONCLUSION AND FUTURE WORKS

This paper proposes a novel solution for real-time indoor map generation for the spatial environment using crowd-sourced data. The primary intention of this research was to come up with a solution using BLE (iBeacons) technology and smartphones which are owned by almost everyone today. Data storage and the processing are done on the server as mobile devices have limitations in processing power and storage. An API was developed to collect data from users. The map will be updated and developed with users' contribution, and users can use the generated map to navigation. This paper describes proposed solutions, methodologies and techniques used and the testing. Relaxation algorithms and error correction algorithms used to increase the accuracy of future work. In addition, data compression methods will be used to reduce the processing time. This system will be further developed to collect different data such as Wi-Fi data, CCTV camera data and data from other IOT devices such as printers so dynamic API implemented to support different data types.

7. ACKNOWLEDGMENT

This work has been supported by the Sri Lanka Institute of Information Technology, Malabe, Sri Lanka'.

8. REFERENCES

- [1] K. K. Naik and M. N. G. Prasad, "A system for locating users of WLAN using dynamic mapping in indoor and outdoor environment-LOIDS," 2008 11th International Conference on Computer and Information Technology, Khulna, 2008, pp. 156-160.
- [2] V.S.P Vidanapathirana, K.H.M.R Peiris and D. Dhammearatchi, "Hospix: The Hospital Exploring Application for Smart Devices," in International Journal of Scientific and Research Publications, vol. 5, no. 11, pp. 489-492, Nov. 2015.
- [3] C. Wen, L. Qin, Q. Zhu, C. Wang and J. Li, "Three-Dimensional Indoor Mobile Mapping with Fusion of Two-Dimensional Laser Scanner and RGB-D Camera Data," in IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 4, pp. 843-847, April 2014.
- [4] J. Aulinas, Y. Petillot, J. Salvi and Xavier Lladó, "The SLAM problem: a survey," in Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, 2008, pp. 363-371.
- [5] J. Du, Y. Ou and W. Sheng, "Improving 3D indoor mapping with motion data," 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, 2012, pp. 489-494.
- [6] C. Wen, L. Qin, Q. Zhu, C. Wang and J. Li, "Three-Dimensional Indoor Mobile Mapping with Fusion of Two-Dimensional Laser Scanner and RGB-D Camera Data," in IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 4, pp. 843-847, April 2014.
- [7] A. Janowski and M. Bednarczyk, "Considerations on Indoor Navigation Based on Cheap Mobile Devices," 2016 Baltic Geodetic Congress (BGC Geomatics), Gdansk, 2016, pp. 78-84.
- [8] Y. Li, Y. Zhuang, H. Lan, P. Zhang, X. Niu and N. El-Sheimy, "Self-Contained Indoor Pedestrian Navigation Using Smartphone Sensors and Magnetic Features," in IEEE Sensors Journal, vol. 16, no. 19, pp. 7173-7182, Oct.1, 2016.
- [9] J. A. D. C. A. Jayakody and I. Murray, "Proposed novel schema design for map generation to assist vision impaired in an indoor navigation environment," 2014 International Conference on Contemporary Computing and Informatics (IC3I), Mysore, 2014, pp. 490-494.
- [10] Developer.estimote.com. (2017). Cite a Website - Cite This for Me. [online] Available at: <http://developer.estimote.com/> [Accessed 17 Aug. 2017].
- [11] Altbeacon.github.io. (2017). Android Beacon Library. [online] Available at: <https://altbeacon.github.io/android-beacon-library/distance-calculations.html> [Accessed 17 Aug. 2017].
- [12] Rodriguez, U., Frisk, F. and Klonowska, K. (2011). Indoor Positioning using Sensor-fusion in Android Devices.
- [13] API, C. (2017). Camera API | Android Developers. [online] Developer.android.com. Available at: <https://developer.android.com/guide/topics/media/camera.html> [Accessed 17 Aug. 2017].
- [14] G. Amat, J. Fernandez, A. Arranz and A. Ramos, "Using Open Street Maps data and tools for indoor mapping in a Smart City scenario", in International Conference on Geographic Information Science, Vienna, Austria, 2014.
- [15] Altbeacon.github.io. (2017). Android Beacon Library. [online] Available at: <http://altbeacon.github.io/android-beacon-library/distance-calculations2.html> [Accessed 17 Aug. 2017].