2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)

Real-Time System for Indoor User Localization and Navigation using Bluetooth Beacons

Ievgen Gorovyi, Alexey Roenko
IT-Jim
Kharkiv, Ukraine
ceo@it-jim.com, a.roienko@it-jim.com

Alexander Pitertsev*, Ievgen Chervonyak*, Vitalii Vovk**
*Electronics Department,
**Aviation radio-electronic complexes Department
National Aviation University
Kiev, Ukraine
pitertsev@gmail.com, eugenecher94@gmail.com

Abstract — **Real-time user positioning and navigation services are widely used daily by millions of people. The challenge is that common global positioning systems fail in indoor environment or in scenes with a limited sky view. In the paper, the indoor navigation framework based on the Bluetooth beacons is proposed. Such system allows user to obtain the position in GPS denied environment on a real-time basis. Key steps for precise localizing and main components of the developed framework are considered. Initial experimental results are provided.**

Keywords - Indoor navigation; Bluetooth; beacon; user localization; trilateration.

I. Introduction

User positioning and navigation systems are widely applied nowadays. Owing to the known position of the device, such systems can help their users to look for the nearest hotels, restaurants, sights or to find the vacant places on the multistoried parking, the shop of the interest in a huge shopping mall or the nearest registration desk at the airport.

At present, positioning and navigation systems that are based on a set of satellites are considered as the most popular ones [1]. This is due to the fact that nowadays almost any ordinary smartphone or tablet has a receiver of signals of global positioning system known as GPS [2]. Recently another system, GLONASS, has become popular as well. It is worth noting the navigation system, Beidou, made by China is intended to start working worldwide in 2020. Despite the fact that such systems can work properly almost in all weather conditions, the main challenge is that there should be a line-of-sight between user's device and at least four satellites [1]. It is seldom satisfied in megalopolises with compact planning. There are even more deterioration factors in this case: reception of the copies of satellite signals reflected from the buildings, signal attenuation, noisy environment [1].

Line-of-sight requirement is definitely lost inside the buildings. Then the possible solution is the application of so-called Indoor Positioning and Navigation Systems (IPNS) [3]. Their implementation discovers such possibilities as: to guide passengers along common navigation paths in airports such as public transport points, check-in counters, immigration, boarding/arrival gates and baggage claim areas; to help patients to reach for the various departments and doctors in hospitals as well as to locate a proper medical equipment for stuff; to find the interesting exhibit at a museum; or to get to the needed department and shelf with the goods in a shopping mall [3, 4].

Various technologies can be used in order to build the IPNS. Among them it is worth noting the geomagnetic-based strategies, technologies employed optical methods, sound propagation techniques, sensor-based methods, radio-based technologies [5]. Each group mentioned has both benefits and drawbacks but their overview is out of the paper's scope. However, it is worth noting that recent years the radio-based IPNS have been of great interest due to the possible accuracy which can be achieved, computational simplicity comparing to other techniques and relative inexpensiveness [3].

Such kind of systems can be designed on the basis of several radio interfaces. The main of them are Wi-Fi and Bluetooth [4]. The idea of Wi-Fi-based IPNS is to capture the signals from active wireless access points and compare them with the signal fingerprints measured for the space in advance. Then, knowing the coordinates of Wi-Fi routers user's location is estimated [4]. Despite of simplicity, the main difficulties of such an approach are the facts that the access points location can be changed any time, rather great error in position estimation (up to 10 m), some software restrictions which are hampered the system implementation [6].

IPNS on the basis of Bluetooth interface utilizes the small autonomous devices known as "beacons" (Fig. 1). They were firstly introduced by Apple in the middle of 2013 and since then have got a great interest all over the world [3]. Beacon itself is a small low-power transmitter based on Bluetooth Low Energy (BLE) technology. It periodically broadcasts data package containing beacon's ID and some additional data. Depending on the transmission power level set and the presence of different obstacles coverage range can be up to 25-40 m. From the theoretical point of view, the positioning algorithm in the case of beacons-based IPNS is rather simple and was described long ago [7]. It states that if at least there are three radio-transmitters and we exactly know their coordinates and output power then measuring the power level of the signals received by user's terminal it is possible to estimate the user's location with respect to the transmitters. The task described is known as trilateration and triangulation problems [7]. Their solutions are based on the fundamental statement of radiolocation theory that the transmitted signal power is diminished reversely proportional to distance squared.

978-1-5090-3006-4/17/$31.00 ©2017 IEEE

Fig. 1. Different models of the beacons.

At present, there are three main industrial interfaces that provide necessary functionality for designing the IPNS based on beacon technology. The first one is known as iBeacon and was introduced in the middle of 2013 by Apple [8]. Later, in 2014, Radius Networks published another protocol called AltBeacon [9]. One year later Google ran its version of interface known as Eddystone, at the same time providing the support for the first two interfaces [10].

Despite of the simplicity of the background idea for locating the device on the basis of beacons, it is occurred to be not so easy task. First of all, the main difficulty has arisen due to the difference in signal propagation outside and inside the buildings. The latter case is characterized by signal reflections from the walls, interference and attenuation due to the different obstacles [11]. As a result, power decay rate of BLE signal usually differs from that applied in solutions for outside radiolocation tasks [12]. Detailed investigations and attempts to cope with these practical problems have led to the development of several interesting techniques and corresponding Software Development Kits (SDK) by some companies (Navigine, Atlas, Estimote, InfSoft, Kontakt.io etc.).

Among them it is worth mentioning a technique known as fingerprinting which addresses the question of user's location by preliminary acquisition of radio imprints all over the place intended for the navigation [3, 13]. Then, while running, system tries to find the best matching of the radio signals received by user's device with the imprints stored using computer science methods at a server part. Despite of the many approaches to design the beacons-based IPNS, the accuracy that can be practically reached is about 1-2 m [3, 11]. Some potentially effective solutions are hidden due to the commercial usage and that is why cannot be analyzed properly. The goal of the paper is to reveal some key design details of the beacons-based IPNS and to represent some operating examples using the working prototype of the system.

The paper is organized as follows. Section II describes the main principles of the IPNS positioning stage and highlights some their technical details. Section III represents the description and contains some examples for navigation part of the system. The design structure of the working IPNS prototype and the acquired results are considered in section IV. Finally, conclusions follow in the last part of the paper.

II. User Localization: Principles And Challenges

A. *Bluetooth Beacons Fundamentals*

There are several key steps that should be considered in context of user indoor positioning task. The first one is the determination of beacons placement. Optimal beacons positions are typically determined empirically, but some recommendations can still be given. In order to avoid the appearance of multiple signal copies, it is necessary to locate beacons as far as possible from metal objects. One of the best solutions is to fix beacons on a ceiling far from the walls in order to provide line-of-sight condition for the greatest number of user's devices. Average distance between beacons may vary from 10m to 50m depending on the module transmitting power set as well as building plan and construction.

The second step is the proper configuration of a beacon's broadcast information. As an example, let us consider a package structure for iBeacon interface (Fig. 2):

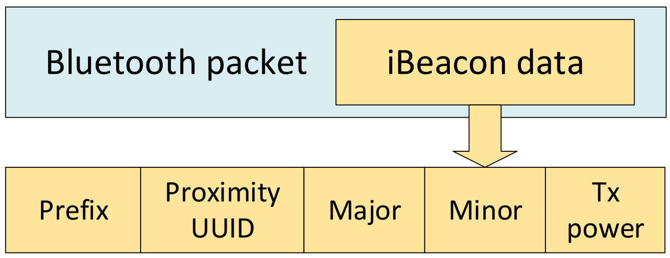

Fig. 2. Package structure for iBeacon interface.

Protocol data unit (PDU) can be considered as a key part of the data package. It consists of several sections, namely, header, MAC address of a beacon and data section. It is worth noting that MAC address is sometimes hidden and cannot be read in some mobile platforms. In such a case, beacons identification and belonging to the corresponding IPNS can be done by setting the unique values for the fields "Proximity UUID" (Universally Unique Identifier), "Major" and "Minor". While the former field usually gives the information about a particular beacons group index, the latter two allow identifying the beacon within a given group.

The last field in the "Data" part of the package, "Tx power" (transmitter power), contains the value of signal power at 1 m distance from the beacon. This value is necessary for calculating the distance to a beacon at a user's device knowing the value of a received signal strength indicator (RSSI) for a beacon package at hand.

RSSI is considered as the main parameter and a basis of IPNS based on BLE beacons. It linearly depends on the received signal power and usually is measured in dBm. Fig. 3 illustrates an examples of RSSI signals from one of the beacons for the situation when user's device is in static position, there is a LOS and manually measured distances are equal to 3 m and 5 m correspondingly. It is clearly seen that there is a strong deviation of RSSI samples with respect to their mean values most probably caused by the interference between direct arrival signal and its copies reflected from the walls.

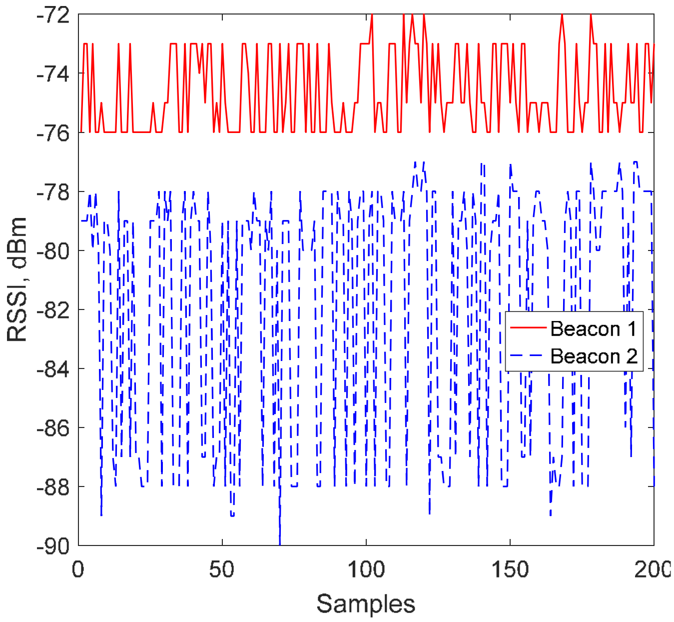

Fig. 3. Example of RSSI values from different beacons.

Theoretically, RSSI fluctuations are divided into two types: low-frequency and high-frequency fluctuations. Fast changes can commonly be compensated using RSSI filtering while low-frequency changes are usually the result of user's movement and should be properly processed [12]. The choice of the filter for suppression of high-frequency fluctuations is considered as the third key step for user's positioning task. The modifications of moving average filter are possible solutions at this stage proposed by AltBeacon protocol [9].

The next key step is the evaluation of the distance between the user and particular beacon. In order to do this, the empirical dependence is typically utilized

$$d(t) = 10^{(RSSI_{d_o} - RSSI(t))/10n}, \quad (1)$$

where $d(t)$ is the distance from beacon to user's terminal at the t moment of time, $RSSI_{d_o}$ is a power at the distance d_0 (commonly $d_0 = 1$ m and $RSSI_{d_o} = Tx_p$), n is a damping factor.

Estimation of n parameter value is a main challenging task at this stage. Such procedure is typically called beacons calibration process. Fig. 4 contains the dependences of the distance on RSSI for different values of damping factor for $RSSI_{d_o} = -65 dBm$. One can see that an inaccurate value of n parameter can lead to significant errors in distance estimation. For example, if the set value of n is 2, but the real value is 3, the error in distance estimation can be more than two times (see the curves values for $RSSI_{d_o} = -85 dBm$). It is also worth noting that the smaller the received RSSI values (i.e. the farther the terminal from the beacon) the greater the influence of wrong damping factor value set. In order to estimate the proper value of n parameter, it is enough to have the value of Tx_{power} in two calibration points with a priori known distances [12, 14].

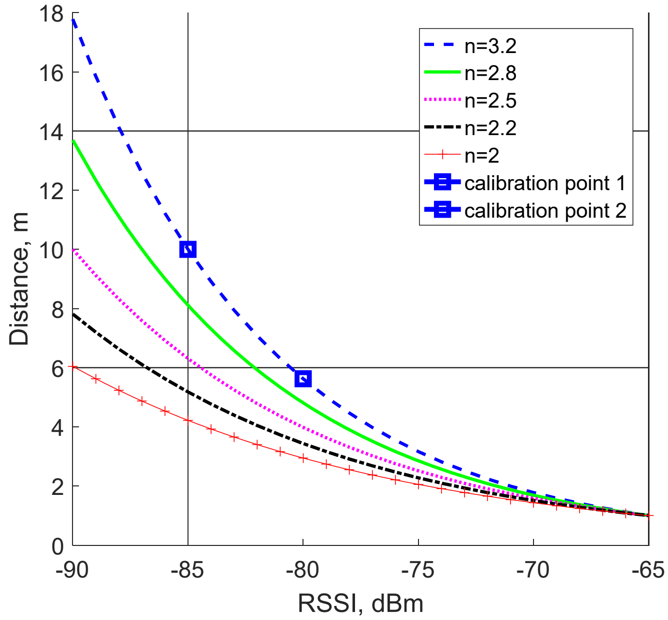

Fig. 4. Dependencies of distance on RSSI values for different values of damping factor.

Usually all required measurements are accomplished during the system deployment by a support engineer (done once). The calibration procedure can be considered as optimization process with respect to a set of beacons parameters

$$f(Tx_p^1, n_1,.., Tx_p^k, n_k) \to \min, \quad (2)$$

where (Tx_p^i, n_i) are target parameters for each beacon. An advanced way to accomplish the calibration is to use a ground truth (user trajectories) and minimize the errors between true and retrieved paths. Experimental section contains real examples demonstrating a substantial effect of RSSI filtering and calibration.

B. Positioning Stage

In order to determine the coordinates of the user's terminal it is enough to have the distances to three beacons, d_i, as well as to know their coordinates (x_i, y_i, z_i) (Fig. 5a). Such practical task is known as trilateration algorithm [15]. The answer can be found by solving the system of the following equations:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i, i \in [1;3], \quad (3)$$

where (x, y, z) are user coordinates.

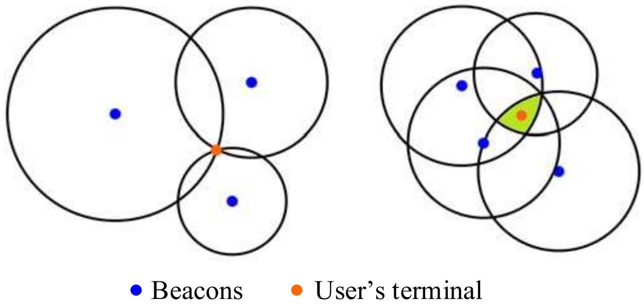

Fig. 5. Trilateration and multilateration diagrams (a – trilateration, b - multilateration).

One clarification should be done at this step. If the goal is estimation of 3d coordinates of the user, four beacons are required. However, in real scenarios it is often enough to consider the problem in 2d.

If the number of available beacons is more than three, equation (3) can be rewritten in the matrix form for an arbitrary number of beacons

$$\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2x_k & -2y_k & -2z_k \end{bmatrix} \cdot \begin{bmatrix} x^2+y^2+z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_2^2 - z_2^2 \\ \vdots \\ d_k^2 - x_k^2 - y_k^2 - z_k^2 \end{bmatrix}, \quad (4)$$

where k is the amount of beacons. Fig. 5b illustrates a high-level view in the case of multilateration. Solution of (4) provides the estimate of user location. Trilateration and multilateration mathematics are analyzed in more detail in [15].

III. System Configuration and Data Preparation

Consider key stages required for initial system setup. Assuming that building map is available, then two steps described above (see Section II) are necessary. The first one, properly locate beacons and store their coordinates with respect to the map coordinate system. Secondly, beacon calibration stage is applied. As a result, each beacon is described by coordinates (x_i, y_i), parameters $RSSI_{d_o}$ and n_i.

In order to provide the possibility for user navigation, the building map should be processed in proper way. Obviously, that map can be divided into two regions corresponding to allowed and not allowed areas for user movement. Fig. 6 illustrates an example of such building map mask. Thus, the map mask is just a binary image (consisting on zeros and ones).

In order to perform the preparation for user navigation, a specific graph structure should be built from the map mask [17]. In the simplest scenario, the graph layout could be created manually. It will be enough to put nodes on the way turns and intersections. However, in the case of large buildings automatic solution is much more preferable. Two alternatives can be considered for this purpose: skeletonization techniques and navigation mesh algorithms.

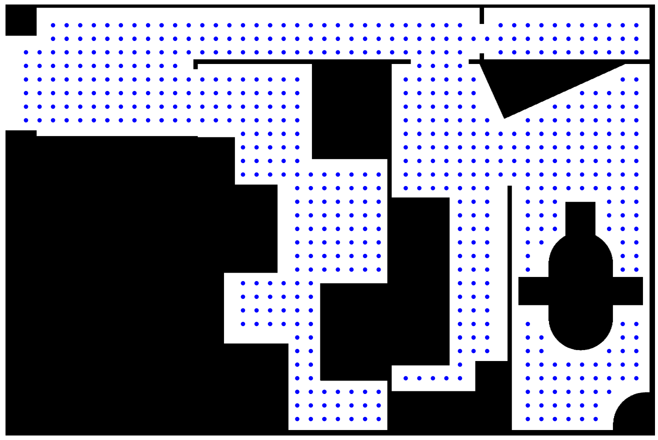

Fig. 6. The mask of the building map (area is about 52 m^2).

The idea of skeletonization is to consequently apply the thinning algorithm while keeping the topology of the object structure the same. Alternatively, a navigation mesh solution can be used for the same purpose [17]. This approach has received significant development in the computer games industry and was described in several publications [17, 18]. Its idea is based on the map division on a sequence of squared tiles. At the next step, we check if all the pixels of the tile belong to the allowed area. Based on such checks the navigation nodes are generated (see dots in Fig. 6). As a result, one can obtain a sequence of graph nodes.

The next step after placing the nodes is to define coefficients of the adjacency matrix. We should check, whether the walls or other obstacles are present in the line of sight between each pair of the nodes. If no obstacles are present, the weight of such edge of the graph will be evaluated as Euclidean distance between the nodes.

As a soon as graph structure is constructed, routing algorithms can be applied for user navigation In terms of routing, the problem is to find the shortest path for a given undirected graph. Dijkstra algorithm and A* can be used for this purpose [17]. Basically, both algorithms use the same input. There is a difference in the optimal path construction. The Dijkstra's algorithm determines the shortest path, alternately visiting each of the neighboring nodes, while A* algorithm relies on the heuristic characteristics of the estimated destination point proximity, and tries to follow this direction. Obviously, that both routing algorithms provide the optimal path and, hence, the same destination point.

IV. Experimental Results

This section contains description of architecture of the built IPNS and initial experimental results.

A. *Key Components of IPNS*

Let's consider the high-level diagram of the developed framework (Fig. 7). Initially, building maps and required configuration data are loaded to the application server by the service provider. This step is accomplished only once. As for the user, he should just install the client application.

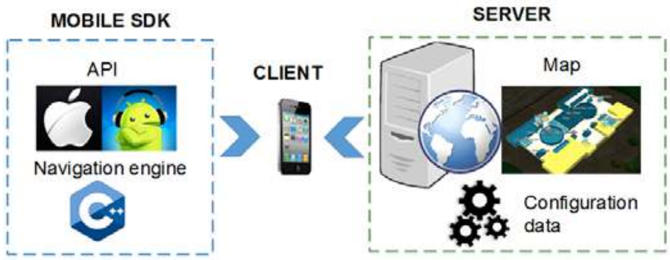

Fig. 7. High-level diagram of indoor navigation system.

This application is connected to the indoor navigation software development kit (SDK). The SDK itself contains two parts: navigation engine and mobile application programming interface (API). The former is a set of algorithmic routines implemented in C++. In particular, positioning algorithms, routing algorithms, signal analysis methods, etc. The latter contains the native Android and iOS code working as a bridge to the navigation engine. One should emphasize, that the configuration data can be downloaded initially during the client application installation. After that the system does not requirei nternet connection and provides user positioning and navigation in real-time basis. All calculations are performed directly on mobile device.

B. *Positioning Accuracy Tests*

Estimation of user's location accuracy comprises a number of experiments including measurements of static user location as well as dynamic cases while user moving along a predetermined trajectory. To reduce the influence of reflected signal copies, we adhere to rules avoiding the beacons installation in the corners and close to walls. The final user position can be calculated using multilateration technique in 2D plane, excluding the relative *Z* coordinates of beacons and receiver (navigation terminal) or in 3D plane. Our experimental tests have shown that in case of small difference in levels between the devices (about 1-2 m with respect to the horizontal distances between them), more accurate results can be obtained in 2D mode.

Fig. 8 contains an experimental example of user positioning in the static case. We have compared the results for different scenarios in order to demonstrate the effect of calibration and preprocessing. One can observe that filtering of RSSI values (triangles) leads to less spread of estimated user locations. And the calibration effect is noticeable providing the positioning precision within 0.5m. It is seen that the real user trajectory and estimated paths with and without appropriate signal processing. It is obvious that beacons calibration significantly improves the system efficiency. The positioning precision is around 1-2m in this case. Example of dynamic experiment is illustrated in Fig. 9.

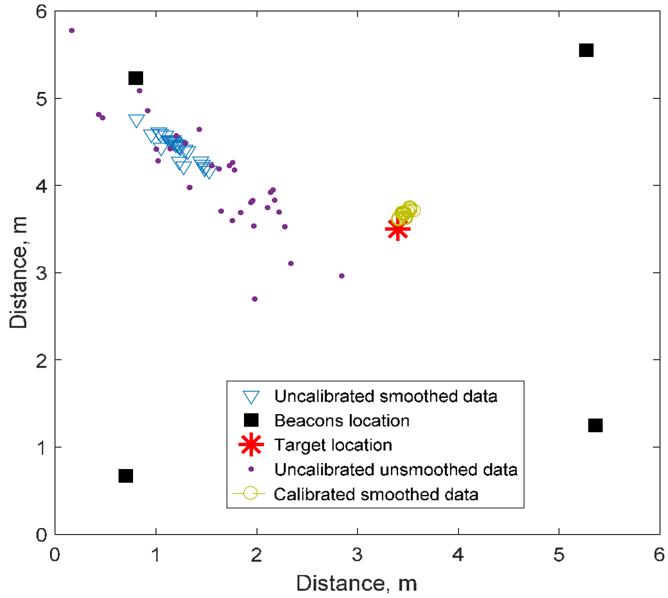

Fig. 8. Static case.

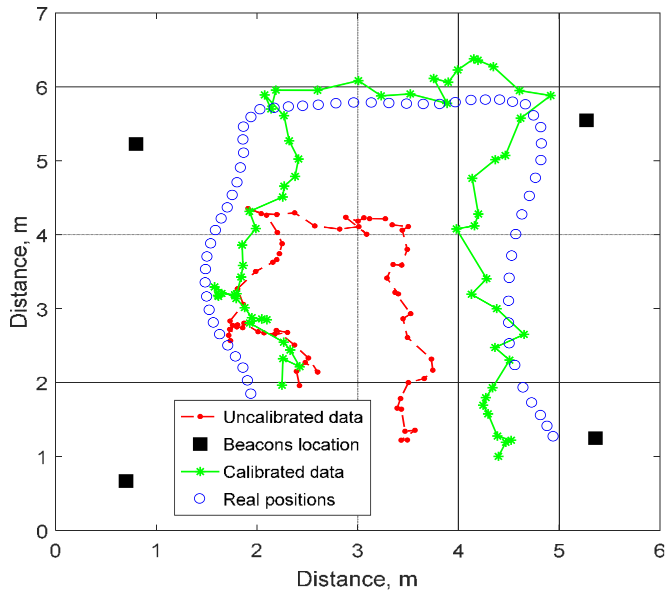

Fig. 9. Dynamic case.

V. Conclusion and Future Work

An indoor navigation system based on Bluetooth beacons is presented. Key system components are described. Peculiarities of data acquisition, filtering, user positioning and navigation are considered.

Experimental results reveal good potential of the framework for real-time user navigation in indoor environments. In the near future, we intend to improve the accuracy of the IPNS via development of more intelligent data processing approaches and integration of the visual methods such as simultaneous localization and mapping technology (SLAM).

VI. References

[1] H.A. Karimi, “Indoor Wayfinding and Navigation”, CRC Press, 2015.

[2] A.K. Mani and V. Agrawal, “Satellite Technology: Principles and Applications”, John Wiley & Sons Ltd., 2014.

[3] H.A. Karimi, “Universal Navigation on Smartphones, Springer Science+Business Media”, 2011.

[4] S. Goswami, “Indoor Location Technologies”, Springer Science+Business Media, 2013.

[5] S.A. Ahson and M. Ilyas, “Location-Based Services Handbook: Applications, Technologies, and Security”, CRC Press, 2011.

[6] H. Dongsoo, J. Sukhoon, L. Minkyu and Y. Giwan, “Building a Practical Wi-Fi-Based Indoor Navigation System”, IEEE Pervasive Computing, vol. 13(2), pp. 72-79, 2014.

[7] C. Venkatramaiah, “Textbook of surveying”, Universities Press, 1996

[8] A. Cavallini, “iBeacons Bible”, 2015, [PDF], p.5, https://meetingofideas.files.wordpress.com/2015/09/beacon-bible-3-0.pdf.

[9] Radius Networks, 2014, https://github.com/AltBeacon/spec

[10] R. Amadeo, “Meet Google’s “Eddystone” - a flexible, open source iBeacon fighter”, Ars Technica, 2015, https://arstechnica.com/gadgets/2015/07/meet-googles-eddystone-a-flexible-open-source-ibeacon-fighter/

[11] C.C. Pu, S.Y. Lim and P.C. Ooi, “Measurement Arrangement for the Estimation of Path Loss Exponent in Wireless Sensor Network”, Proc. of the 7th Internat. Conf. "Computing and Convergence Technology (ICCCT)", 2012, pp. 807-812.

[12] D. Qian and D. Waltenegus, “Evaluation of the reliability of RSSI for Indoor Localization”, Proc. of the 7th Internat. Conf. “Wireless Communications in Unusual and Confined Areas”, 2012.

[13] M. Wang, “Indoor Navigation Systems Based On iBeacon Fingerprinting”, Thesis for the degree of Master Of Science, Nashville, TN, 2015.

[14] M. Matteuchi, “An adaptive indoor positioning system based on Bluetooth Low Energy RSSI”, Politecnico di Milano, 2012/2013, pp. 31-33.

[15] A. Norrdine, “An Algebraic Solution to the Multilateration Problem”, Proc. of the 3rd Internat. Condf. “Indoor Positioning and Indoor Navigation”, Sydney, Australia, 2012.

[16] A. Foerster and A. Foerster, “Emerging Communications for Wireless Sensor Networks”, InTech, 2011.

[17] S. Brand, “Efficient obstacle avoidance using autonomously generated navigation meshes, Master thesis, Delft University of Technology”, 2009.

[18] Y. Bjornsson, M. Enzenberger, R C. Holte and J. Schaeffer, “Fringe Search: Beating A* at Pathfinding on Game Maps”, CIG, 2005, pp. 125-132.