Indoor Mapping and Positioning using Augmented Reality

İbrahim Alper Koç[1], Tacha Serif[1], Sezer Gören[1], George Ghinea[2,3]
[1]Dept. of Computer Engineering, Yeditepe University, Ataşehir 34755, Istanbul, Turkey
[2]Dept. of Computer Science, Brunel University London, Uxbridge, UB8 3PH, Middlesex, UK
[3]Dept. of Technology, Kristiania University College, 0186 Oslo, Norway
[1]alper.koc@persystlab.org, [1]tserif@cse.yeditepe.edu.tr,
[1]sgoren@cse.yeditepe.edu.tr, [2,3]george.ghinea@brunel.ac.uk

***Abstract*— Location-based services are becoming an important part of life and there is an increasing demand for indoor positioning. Combination of GPS and mobile cellular networks has solved the problem for outdoor environments. However, the same level of precision has not been achieved for indoor location estimation. The problem of locating an indoor environment has been studied only recently. Much research contributed to the innovative concept of an indoor positioning system. Considering the cost and the effort involved in the existing location estimation approaches, Augmented Reality (AR) based positioning method is one of the most promising methods to determine the location of a mobile device. Accordingly, in this paper, we propose, implement, and evaluate an AR-based location estimation smartphone application that can be used indoors. The evaluation results show indicate that the proposed application can estimate the location in small areas with an error-rate of 0.31 meters and in large areas with error-rate of 7.8 meters.)**

Keywords— Augmented Reality, Mapping, Location Detection, Indoor Positioning, Beacon.

I. INTRODUCTION

Global Positioning System (GPS) is designed to detect the location in outdoor environments where there is line-of-sight between the satellites and the receiver. Even though there has been a considerable amount of studies undertaken in indoor positioning, none of the proposed solutions were nearly as comprehensive as the GPS.

On the other hand, in the past decade, with the introduction of smartphones, there have been changes in people's everyday life. Nowadays, companies on top of their online and mobile commerce outlets, they are utilizing customer mobile devices for pushing information or even special offers based on the customer's location. However, even though with aiding tools such as accelerometers, magnetometers, gyroscopes, and Bluetooth trans-receivers, it is not possible for a smartphone to locate itself in an indoor environment. Hence, various techniques and technologies have been proposed for indoor location estimation. Most of these techniques are exploiting the mobile device's existing wireless technologies such as Wi-Fi 802.11 [1], Bluetooth [2] and some others are utilizing its internal sensors [3].

Bearing in mind the existing solution location estimation methods, a novel approach, which does not put wireless technologies in its core, has been proposed in this paper. This approach involves the use of Augmented Reality (AR) for indoor environment mapping which then coupled with internal sensors of the mobile device can be used to track the device's movements.

Accordingly, this paper is structured as follows: Section 2 highlights similar work in the literature. Section 3 details the methodologies and hardware/software tools that are required to implement the targeted system. Section 4 elaborates on the requirements and the design of the proposed solution. Section 5 describes the steps undertaken to implement the proposed system. Accordingly, Section 6 presents the evaluation set up and presents the findings of the evaluations conducted. Last but not least, Section 7 concludes the work that is undertaken and reviews possible future work venues.

II. BACKGROUND

Hightower and Borriello [4] proposed the taxonomy of location systems for mobile-computing applications in a survey paper. In [4], 15 different location-aware applications were compared according to accuracy, precision, scale, costs, and limitations. One of the mentioned location systems in [4] is the RADAR project. The RADAR project [5] by Bahl and Padmanabhan was one of the early examples of location-aware systems and a significant contribution to the field. RADAR project is one of the first studies that had used the radio signal to locate and track user within a building. The researchers in this project aim to estimate the location of the user by processing signal strength from base stations that are distributed in the test bed area. The tests in this study are conducted in an area of 980 square meters and by utilizing three wireless adapters equipped base stations. Their findings indicate that median resolution of the RADAR system was in the range of 2 to 3 meters, which as it has been stated by the authors it can be used to implement an interesting class of location-aware services, such as printing to the nearest printer, navigating through a building.

Similarly, another trendsetter study which has utilized WiFi fingerprinting for indoor location detection was conducted by Farshad et al. [6]. In this study, authors explore the impact of various aspects underlying a WiFi fingerprinting system, which includes the definition of a fingerprint, run-time location estimation algorithms, frequency band and presence

978-1-7281-2888-7/19/$31.00 ©2019 IEEE
DOI 10.1109/FiCloud.2019.00056

of virtual access points. Their investigation considers several different real indoor environments ranging from a multi-story office building to shopping centers of different sizes. In this study, WiFi fingerprinting data is obtained using the Android device using IndoorScanner application, which scans WiFi signals 20 times in a second and records all RSSI data to the database using MySQL. Their findings indicate that the combination of fingerprint definition and location estimation algorithm that yields the best location accuracy is highly dependent on the environment and even specific floor within a given environment.

On the other hand, the study undertaken by Kim and Jun [7] proposes an indoor positioning technique based on AR algorithms. The system proposed by the authors automatically recognizes a location from image sequences taken from indoor environments. The image sequences are recorded by a wearable mobile PC with a camera which captures and transmits them to the image database. This database holds visual information about indoor environment locations and paths between these locations. To recognize a location, a pre-constructed image database and location model is utilized. According to the evaluations conducted by the authors, the proposed system achieved an average location recognition success rate of 89%.

The idea of combining location and augmented reality has been around for some time. One the first examples of these studies have been conducted by Paucher and Turk [8], who proposed a new approach for indoor location detection and pose estimation so that can be used as a supporting framework for augmented reality applications on mobile phones. In this study, the authors utilize the mobile device's embedded camera and sensors to localize the device in a familiar environment and determine its orientation. Their tests results, which were conducted in the university offices, indicate that the projection error is about four pixels for a 640×480 image for the augmented picture and localization error is around 10 to 15 centimeters.

Similarly, Ahn and Han [9] have developed an indoor evacuation system, which makes use of personalized pedometry, to guide users in distress out of a building by the help of augmented on-screen elements (Fig 1.). The proposed solution was implemented on a mobile device running Android 2.3 (Froyo) operating system. The proposed system is composed of four components; these are (a) pedometry-based localization; (b) evacuation path recommendation; (c) augmented reality and lastly (d) image-based positioning. These four components build up the most crucial services provided by the application. The pedometry-based localization component bearing in mind the users heading calculates the steps taken by the user and estimates user's location within the building. Evacuation path recommender is a smart mapping tool that takes into consideration user's walking speed, his/her current location and the nearest exist and proposes an evacuation path. The augmented reality component using user data and the pre-offered evacuation path recommendation draws lines and arrows on the screen so the user can easily follow. Lastly, the image-based positioning component improves and refines the location detection process that was undertaken using pedestrian dead reckoning by the help of a commercial image tagging web service, which helps the recognition of a specific location using images captured by the phone camera.

Fig.1 Emergency guide system using augmented reality [9]

Accordingly, the step count accuracy was tested with three different sized (Short, Medium and Long) strides. Their overall findings indicated that the average error rate in detecting user steps was 3.33 percent. However, the second test which was on map matching, considering the user's heading and orientation, deviated and provided erroneous results.

Similar to the previous project, CAViAR (Context-Aware Visual indoor Augmented Reality for a University Campus) aims to develop an augmented reality enabled indoor navigation system which is part of a larger intelligent campus environment project [10]. The proposed system is implemented in the iOS environment. However, unlike the previous work, it cannot navigate users within a building but can do voice search and show their possible destination on a map. Furthermore, combining mobile devices sensors and a commercial image tagging software development kit, it can identify user's location and provide information about its current surroundings (Fig 2). The information is provided as an overlay to the real-time camera using OpenGL ES augmented reality tools.

Fig.2 CAViAR augmented reality campus guide [10]

The proposed system evaluation showed that, when a user strictly holding the phone in horizon position, the error rate after walking 300 ft was around 2%. However, since the proposed solution was utilizing the phone's sensors and using inertia navigation is very much error prone and dependent on the users sudden or soft movements.

III. Tools and Methodologies

This section details possible tools and methodologies that can be used to implement the proposed system. Initially, it details an augmented reality library that can be used to model the real world. As a follow-up, the methodologies that can be used for localization are described and detailed.

A. ARKit Framework

Augmented Reality Kit [12] (ARKit), is a framework that combines iOS device camera and motion sensors to produce an augmented reality experience. ARKit uses the same analogy – virtual world and camera coordinate system – for all AR applications. This analogy follows a right-handed convention where the y-axis points upward, the z-axis points the observer and the x-axis points to the observer's right side.

Each AR session has its own configuration. AR configuration is provided by setting the position and the orientation of the origin of an AR coordinate system, which respects to the real world.

Using a TrueDepth technology-enabled camera, AR apps on the mobile device can identify the nearby objects of the user's face with the position and the geometry with 3D information. Visual Inertial Odometry [13] (VIO) is the technology, which is used by ARKit to track the nearby real world accurately. VIO combines Core Motion data and Camera Sensor. ARKit senses the movement speed, position and the orientation of the device in the room using the data from these sensors. Results from the VOI have a high degree of accuracy.

The mobile device can analyze the ARWorld scene presented by the camera view and can detect horizontal and vertical planes at nearby space. Objects can be tracked with feature points, with the help of this feature points, new objects can be placed to the ARWorld. ARKit can estimate the total amount of the light in the room and it applies the same amount of the light to the placed objects accurately.

B. Triangulation

Triangulation is a technique of determining the location of an object, based on geometric and mathematical properties of triangles. It uses angles to determine the position of an object. In a 2- dimensional Cartesian coordinate system, two angles and the distance between the reference points can be used to determine the position of an unknown point by the laws of trigonometry. [14]

C. Trilateration

Trilateration is a method that involves using the known positions of three transmitters and their calculated distances to a point to estimate its position. In other words, trilateration is a mathematical technique, in which the location of a point in space is calculated using the intersection of at least three spheres with their given radii and center points. Accordingly, in 2-dimensional space, knowledge of three intersecting and non-identical circles can be used to find a unique solution to the localization problem [7].

Similarly, applications that utilize trilateration usually determine the radii of spheres, or more often circles, using the radio signals propagating from a transmitter with a known location. Assuming uniform signal attenuation and interference-free medium, radio signals collected from at least three known transmitters can be used to localize an object in 2 or 3-dimensional space.

Trilateration involves using distances to identify a point that known to be at the intersection of three or more sphere and circles. In other words, trilateration is a mathematical technique in which the location of a point in space is calculated using surfaces of several spheres with given radii and centers. Accordingly, in 2-dimensional space, knowledge of three intersecting and non-identical circles can be used to find a unique solution to the localization problem [7].

Similarly, applications that utilize trilateration usually determine the radii of spheres, or more often circles, using the propagating signals from a radio device that a known point in space. By meshing up signals from three known radio devices can be then used to localize an object in a 2 or 3-dimensional space.

D. Fingerprinting

Fingerprinting is an algorithm that constructs a map based on nearby signal strength. Received signals have a different RSSI value at every point. At first, reference points should be determined and RSSI data for all the reference points inserted to the database. RSSI mapping phases is completed after these two steps are completed [8].

The fingerprinting method has two phases; the first phase is known as the offline phase or the calibration phase. During this phase, reference points are recorded by populating the radio map with scan results. The second phase is known as the online phase or location detection phase. User's device receives multiple signals in the known indoor environment. Measurements made at this phase are compared with the nearest offline fingerprint to determine the position of the user.

IV. Proposed Application

We propose a novel indoor location detection application based on AR. Our application works on only iOS installed devices. ARKit was used as the AR framework, Swift as the programming language, and Xcode as the IDE. iPhone 6S and newer versions have AR support and all iOS devices have camera, accelerometer, magnetometer, and gyroscope sensors to support AR experience. The iOS device also should have at least iOS version 11 installed. In addition, two beacons are used.

The main steps involved in our approach are as the following:

- ARWorld initialization
- Map construction
- Position detection
- Route construction

A. ARWorld Initialization

In this step, the user's device initiates virtual 3D coordinate system as shown in Fig. 2. User's initial position is the origin of this coordinate system. Both the movement of the user and the distance (as AR coordinates) from nearby objects can be detected by the sensors and the camera of the iOS device.

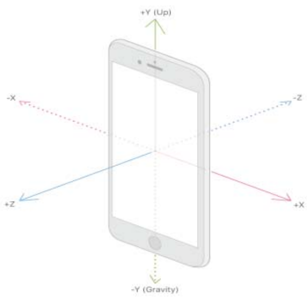

Fig.2 ARKit Coordinate System [12]

B. Map Construction

To construct the map of an indoor environment, the user must mark every corner of the room one by one using ARKit and device sensors. The sequence diagram for the map construction is presented in Fig. 3. When the user starts the application, ARKit generates an ARWorld and starts polling device sensors and the camera. When the user marks a corner of the environment, ARKit calculates its corresponding 3D coordinate. After all, corners are marked, the ARWorld data are stored on the device. The map of the environment becomes ready after marking the fourth corner, but marking more corners instantly updates the current map. Note that all the marked points should have the same height for a more accurate result. Hence, the AR coordinate system provides three values, x, y, and z as shown in Fig. 2. By omitting the y-axis, the 3D coordinate system converts into a 2D coordinate system if and only if all the marked points are at the same height (y values).

Some of the selected points can be placed in the negative zone. All the points must be placed in the positive zone because the iOS screen coordinate system requires positive values. In order to have all the marked points in the positive region, the point with the largest negative x or y value is chosen. The difference from the origin is calculated, and then this amount is added to the corresponding dimension of all points. Consider a 5x5 meter room. The initial (x,y,z) coordinate of the iOS device is (0,0,0). When the ARKit is started to mark the corners of the room, suppose the coordinates of the corners are measured as (2.5, 0, 2.5), (2.5, 0, -2.5), (-2.5, 0, 2.5), and (-2.5, 0, -2.5). Since all the measurements are taken at the same height, we can discard the y-axis values. Considering only (x, z) values, we move all the points to the positive zone so that we now have (5, 5), (5, 0), (0,5), and (0,0). Obviously, the iOS device's coordinate is now (2.5, 2.5). At the end of this operation, all the points are in the positive region. After this, map scaling and rotation operations are applied.

In order to find the scaling factor for map scaling, the two most distant corners of the room are identified, and then the scaling factor is calculated as the distance between these corners divided by the width of the screen. For example, if the width of the screen is 750 pixels, if we leave 50-pixel margin from the left and right sides, then the scaling factor becomes 650/ (longest distance between two points). After scaling the map, then it is rotated until all the points are visible on the screen.

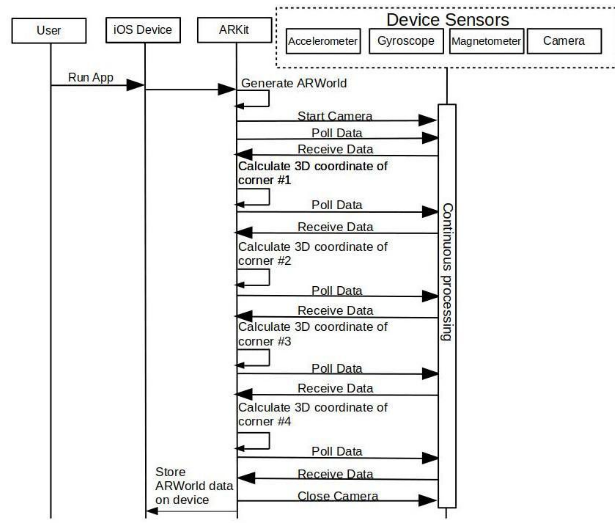

Fig. 3 Sequence Diagram for Map Construction.

C. Position Detection

Fig. 4 depicts the sequence diagram of the position detection step of our system. As shown in Fig. 4, firstly, ARKit is initialized and secondly, ARWorld data are checked for on device. If an ARWorld is constructed before, its data are loaded. Then, the ARWorld is generated and ARKit starts polling the sensors. Based on the data received from the sensors, the user's coordinate is recalculated. In the previous subsection, some operations were performed on the coordinates of the corners during map creation. The user's coordinate is also known on the same coordinate system with the marked corners. The user's coordinates on the map can be recalculated by applying the same operations which are applied to the coordinates of the corners. These operations are repeated for every 0.1 seconds so that the user's location is updated and displayed on the map.

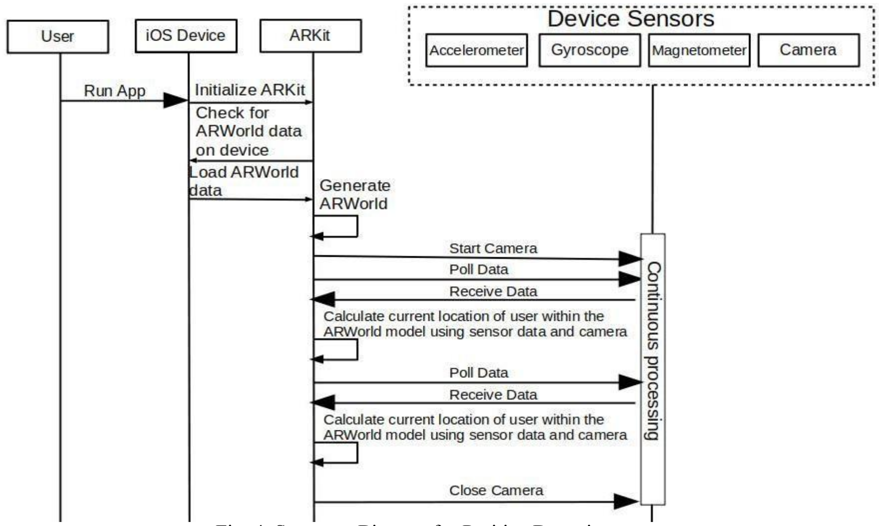

Fig. 4. Sequence Diagram for Position Detection.

D. Route Construction

Once the map construction and positioning have been accomplished successfully, route construction can be started. Two beacons are installed in the mapped area. The first beacon is placed at the starting point and the other one is placed at the end point. The user has to be close to the first beacon in order to start route construction. After this step, the route is drawn as the user navigates through the map. The route is drawn until the user reaches the second beacon and the route construction is completed at that point.

The value of RSSI is checked to see if the beacon is nearby. If the value is greater than -45, the beacon is known as nearby. For smaller values, the beacon is in range but not very close. When the first beacon is nearby, a timer with a 0.1-second interval starts to work. The method launched by the timer is responsible for constructing the route and updating the route every 0.1 seconds. The locations found in each 0.1 seconds time interval is recorded and new ones are added as time progresses. When the second beacon is nearby, the timer stops, and the route is completed.

V. TEST AND EVALUATION

The implemented prototype augmented reality supported mapping and location estimation mobile app is evaluated in the Computer Engineering department's corridors and laboratories. The floor plan of the department, the south wing of the engineering building, is provided in Fig 5.

In order to evaluate the ease of the application of map creation and location estimation precision, two different test environments were selected. The first one was a small area which, assumingly, would be easier to map and identify the location of the user. The second one was a larger area which would mean more effort to construct the digital environment map and to locate the user. Accordingly, six participants were asked to construct the maps and follow the path on the screen. During these processes, all actions were monitored and logged.

Fig. 5. Test area floor plan.

A. Small Area Evaluation

The small test area experiments were undertaken in a small rectangular 9.2x11.9-m^2 room – the Pervasive Systems meeting room. Even though they were not previously ordered so, all participants marked the corners of the room with the same order while constructing the location map. The authors believe that this might be due to the experiment starting point – at the entrance door of the room - which was the same for all participants. For location estimation purposes the participants were asked to walk the same path at a normal pace. There were three reference points identified in the room to make the comparisons. Every time a participant had physically reached to a reference point, his/her estimated location was saved on the mobile device – see Fig. 6. Accordingly, using the reference point coordinates in real-world and the saved estimated location data, the mobile applications estimation error rate was calculated. The Fig. 7 presents the distance error rates for every reference point.

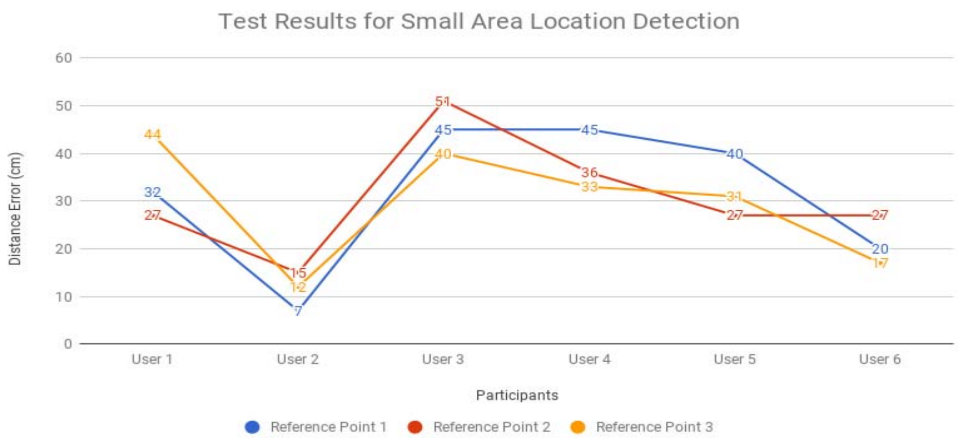

Fig. 7. Test Results for Position Detection in Small Area.

B. *Large Area Evaluation*

The large test area experiment was conducted in the part of the engineering building which is shown in Fig. 5. On the figure, the starting point of the route is indicated with a green diamond symbol, whereas the BLE beacons are indicated with yellow squares. The red line in the figure illustrates the test path, where the three reference points depicted with blue circles. The reference points were strategically selected. Accordingly, the l first reference point was to test the straight path movement tracking capabilities of the mobile application. On the other hand, the second reference point was selected to evaluate the left and right turn detection capabilities of the application. Finally, the third reference point was chosen to observe the overall cumulative error at the end of the path.

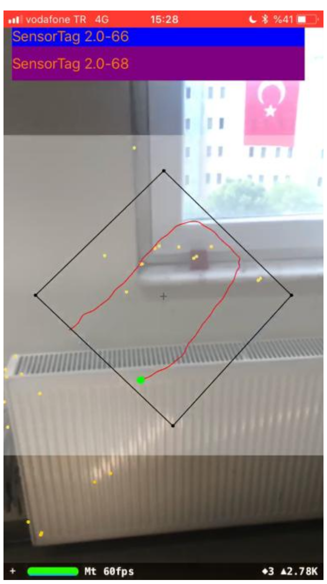

Fig. 6. Small Test Area experiment screen capture.

As opposed to the small test area evaluation, in this test, there were more than 4 mapped corners. Also, this specific corridor was chosen due to its irregular and curvy shape, which can be seen in Fig. 5. The overall area that the experiments took place was 250 square meters. During the evaluations, all the participants strictly followed the same path. At the beginning of their experiment, the participants were asked to construct a map of the corridor. During the map creation, they were asked to follow the same order when they were making the corners of the corridor. After the map of the environment is created, the participants were asked to walk to the end of the corridor following the screen of the mobile device. They were not given any additional information or guidance except on how to use and the mobile application. For every participant, movement data were collected at three specific points – the three reference points. All the participants were asked to walk at a normal pace. Their speed during their evaluation and they were asked to reduce speed if they were walking too fast. The participants were required to follow a bird's eye perspective map of the vicinity, where an example screen capture of the mobile application can be seen in Fig.8.

The location estimation findings of the large area test are presented in Fig. 9. As can be seen from the graph the results for the location estimation in straight movements (Reference Point 1 and 2) are quite promising. However, the error rate gets substantially high at the end of the path (Reference Point 3), which is due to the overall accumulated error. The following figure (Fig.10) indicates the average distance error for the six participants. This graph indicates that the average distance error is around 5 meters and increases while the walking distance is increasing – the highest average distance error rate being 7.8 meters at the 3rd Reference Point.

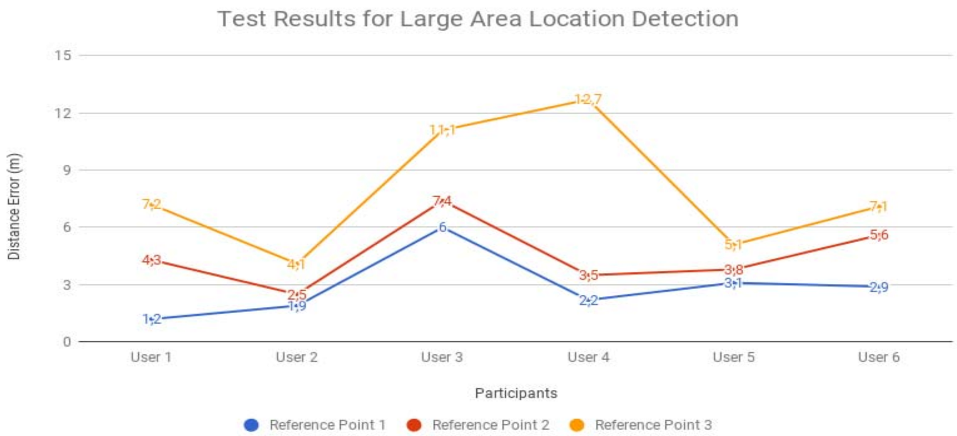

Fig. 9. Test Results for Position Detection in Large Area

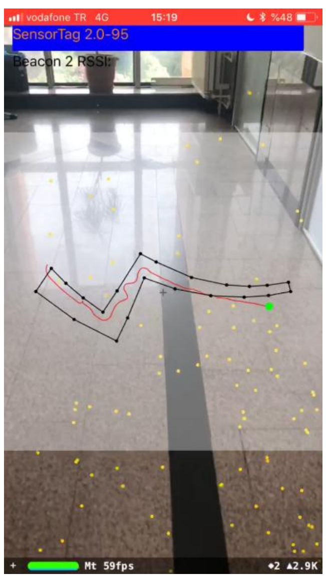

Fig. 8. Large Test Area experiment screen capture.

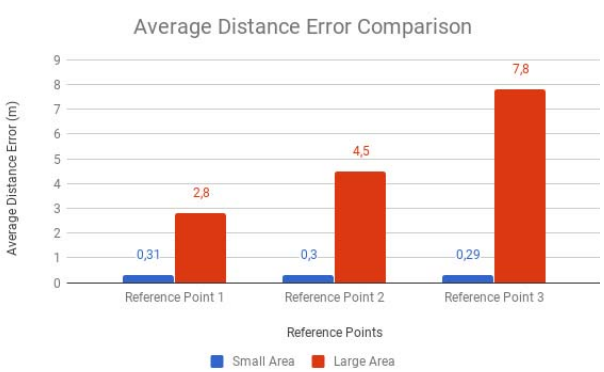

Fig. 10. Average Distance Error Comparisons.

VI. CONCLUSION

The initially proposed mobile system that would enable users to create their own indoor map and location detection system is successfully accomplished. The prototype mobile application can create a map of the vicinity and detect a rough estimate of the location using solely the ARKit framework. However, to improve the location estimation precision, in this work, the mobile app has been coupled with two wireless Bluetooth (BLE) devices that emit signals.

There are a couple of findings highlighted based on the result of the evaluations. Due to lack of familiarity, when users were asked to navigate using a mobile device indicating their path and current location on a bird's view (see Fig. 8) diagram, were struggling. Authors believe that this is due to the familiarity and their expectancy of a perspective view, which is widely used by mobile navigation apps and devices. Furthermore, our findings indicate that even though the ARKit can track user movements and try to detect his/her location using a camera and on-device sensors in small and condensed areas, it requires additional location estimation support employing methods such as triangulation, trilateration and fingerprinting. If the ARKit-based mobile application is used in the larger area it accumulates error in time and as a result, increases the location estimation error rate.

Hence, as future work, multiple signal emitting (Bluetooth Low Energy and/or WiFi 802.11) devices can be integrated to the system which then can be used to detect and correct any location estimation inconsistencies. Also, the shortest path algorithm can be implemented and integrated into the prototype system, which then it will be able to select and offer different paths to the destination point.

REFERENCES

[1] B. Altintas, and T. Serif, "Improving RSS-based indoor positioning algorithm via k-means clustering." In 17th European Wireless 2011-Sustainable Wireless Technologies, pp. 1-5. VDE, 2011.

[2] M.E. Rida, F. Liu, Y. Jadi, A.A.A Algawhari, and A. Askourih, "Indoor location position based on Bluetooth signal strength." In 2015 2nd

International Conference on Information Science and Control Engineering, pp. 769-773. IEEE, 2015.

[3] W. Kang, and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization". IEEE Sensors Journal, vol. 15, no. 5, pp.2906-2916, 2015.

[4] J. Hightower and G. Borriello, "Location systems for ubiquitous computing", *Computer*, vol. 34, no.8, pp. 57- 66, 2001.

[5] P. Bahl, V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System", *Proc. IEEE Infocom* 2000, pp. 775-784, 2000.

[6] A. Farshad, J. Li, M.K. Marina, and F.J. Garcia, "A Microscopic Look at WiFi Fingerprinting for Indoor Mobile Phone Localization in Diverse Environments", International Conference on Indoor Positioning and Indoor Navigation, 2013.

[7] J. Kim, H. Jun, "Vision-Based Location Positioning using Augmented Reality for Indoor Navigation", IEEE Transactions on Consumer Electronics, vol. 54, no. 3, 2008.

[8] R. Paucher, and M. Turk, "Location-based augmented reality on mobile phones." In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 9-16. IEEE, 2010.

[9] J. Ahn and R. Han, "Rescueme: An indoor mobile augmented-reality evacuation system by personalized pedometry", In 2011 IEEE Asia-Pacific Services Computing Conference, pp. 70-77, 2011.

[10] B.A. Delail, L. Weruaga, and M.J. Zemerly, "CAViAR: Context-aware visual indoor augmented reality for a university campus." In Proceedings of The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 286-290. IEEE Computer Society, 2012.

[11] P. Cotera, M. Velazquez, D. Cruz, L. Medina, and M. Bandala, "Indoor robot positioning using an enhanced trilateration algorithm", International Journal of Advanced Robotic Systems, vol. 13, no. 3, pp.110-117, 2016.

[12] Apple, "Augmented Reality – Resources – Apple Developer", Available: https://developer.apple.com/augmented-reality/resources/ [Accessed June 20, 2019].

[13] N. Rajagopal, J. Miller, K.K.R Kumar, A. Luong, and A. Rowe, "Welcome to my world: demystifying multi-user AR with the cloud: demo abstract". In the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 146-147, IEEE Press.

[14] G. Gonçalo and S. Helena, "A novel approach to indoor location systems using propagation models in WSNs", International Journal on Advances in Networks and Services, vol. 2, no. 4, pp. 251-260, 2009.