Wi-Fi Based Indoor Positioning and Navigation System (IPS/INS)

Jaren Golenbiewski
Electrical and Computer Engineering
Kettering University
Flint, USA
jarengski@gmail.com

Girma Tewolde
Electrical and Computer Engineering
Kettering University
Flint, USA
gtewolde@kettering.edu

***Abstract*— Over the last several years there has been fast growing research activities in the areas of indoor positioning, tracking, navigation, and their applications. Commercial, industrial, and retail businesses are highly motivated in developing solutions for accurate localization of assets and people. This paper first presents a brief review of the various technologies for indoor positioning applications, and then chooses one of them, the Wi-Fi RSSI based technique, for further development of a prototype application at a university campus. The algorithms that are commonly used for calculating the user's position are also presented, including Trilateration, Fingerprinting, and the K-Nearest-Neighbor (KNN). For the prototype implementation, an indoor positioning and navigation system is developed with an Android app that any user can utilize to navigate through the target indoor environment. At the end, the paper presents evaluation results of the prototype implementation, which demonstrates the effectiveness of this developed low-cost solution for indoor positioning and navigation applications.**

Keywords*— *Fingerprinting, Indoor Navigation System (INS), Indoor Positioning System (IPS),* K*-Nearest-Neighbor (KNN), Trilateration, Wi-Fi

I. INTRODUCTION

With the proliferation of Smartphones over the last dozen of years, GPS based positioning, navigation, and tracking services have become ubiquitous. The satellite based GPS system works well for the most part in outdoor environments, however, within indoor spaces, GPS signals may be unavailable or become too weak to be usable. It is therefore impossible to rely on GPS for navigation in indoor environments, where people spend most of their time to work and live. Shopping centers, airports, schools, university campuses, hospitals, and museums are just some examples where indoor positioning would be able to bring much needed benefit to people.

Users would like to have access, on their mobile devices, to indoor maps marked with their current location and navigational services that facilitate seamless movement to specific locations of their interest. This technology definitely revolutionizes indoor navigations. Such services could also bring additional benefits to organizations and businesses, from delivery of location-aware services to the offering of location-triggered content and location-based targeted advertising.

Similar to satellite based systems that provide position references for enabling GPS receiver devices to calculate locations, most indoor positioning technologies today rely on some form of beacon sources to serve as anchor points for the localization algorithms. The beacon sources would typically rely on one or a combination of the following technologies for providing reference signals: Bluetooth Low Energy (BLE), Wi-Fi, ultrasonic, infra-red (IR), Ultra-Wideband (UWB), Visual Light Communication (VLC), and Geomagnetic signals.

The rest of this paper is organized as follow. Section II presents the technologies that are used in most of the current implementations of the indoor positioning and navigation applications. The prototype development and implementation of a Wi-Fi RSSI (Received Signal Strength Indicator) based system for indoor positioning in our department is presented in Section III. Section IV presents the algorithm developed for path planning to guide users navigate from their initial position to a destination point. Analysis of the project results are presented in Section V. Finally, Section VI presents conclusions of the paper.

II. INDOOR POSITIONING TECHNOLOGIES

This section presents a brief review of the technologies which are commonly used for indoor positioning systems (IPS) and indoor navigation systems (INS). The complete study with analysis of the different technologies is available in the authors' previous paper [1].

A. Bluetooth Low Energy (BLE)

A BLE beacon is a small low-power device that uses radio signals to broadcast advertising packets at customizable intervals to other devices in the surrounding. These packets can be received by a smart mobile device and analyzed to calculate the approximate distance between the transmitting and receiving devices. If packets from three or more BLE beacons are received by the user's device, the distance information can be used by a trilateration technique to calculate the user's location. This technique requires that the exact placement of the BLE beacons is known, which is leveraged in the position calculation.

Two major categories of beacons, namely iBeacon [2] and Eddystone [3] are currently popular in the market. These BLE

978-1-7281-9615-2/20/$31.00 ©2020 IEEE

beacons utilize software SDKs to extract the radio signal strength indicator (RSSI) measurements for the calculation of distance estimates [4].

B. *Wi-Fi*

Wi-Fi access points (APs) can be utilized as beacon sources in a similar way as BLE beacons for indoor localization. There are a few approaches for utilizing Wi-Fi received signal strength indicator (RSSI) values to estimate the user location. Since each technique has its own benefits and drawbacks, it could be difficult to identify the best technique for a given indoor environment. Typically, choices are made based on the environment configuration as well as the application use case.

1) Trilateration

A trilateration technique using Wi-Fi RSSI measurements is similar to that of the BLE beacon based implementation, except that instead of BLE beacons, Wi-Fi APs are used as beacon signal sources. After receiving RSSI values from three nearby APs, the user's smart mobile device can convert the RSSI values into distance estimates using an RF signal propagation model. It can then apply the trilateration technique which is based on simple distance calculations to determine the user's location. A detailed explanation of the trilateration algorithm is available in [5].

The trilateration technique assumes that the distance measurements are accurate, however, in reality the RSSI to distance conversion, which follows a log-distance path loss model [5] is prone to errors, because of the non-linear relationship between the two and a number of factors that influence the signal strength of the Wi-Fi signal, such as multi-path propagation and interference by other radio signal sources sharing the same frequency spectrum in the environment. Therefore, the trilateration technique may not offer the best position accuracy because obtaining accurate distances from the RSSI measurements is more complicated than it seems.

2) Fingerprinting

In the fingerprinting technique, the incoming RSSI values of the Wi-Fi access points are utilized for positioning by comparing the measurements at the current location with stored RSSI data collected at known reference points. It utilizes statistical relationships to come up with a good estimate of the user's position.

This fingerprinting based positionining technique consists of two phases: training and positioning. The objective of the training phase is to build a fingerprint database for the operating environment. To generate a database, a select set of reference points (RPs) must first be carefully identified to provide sufficient coverage of the operating environment. A mobile user then moves to those selected RP locations one at a time and collects the RSSI values from each AP. This measurement is stored in the database to provide the fingerprint dataset.

During runtime, in the positioning phase, as the user navigates through the environment the user's mobile device collects the RSSI values from its nearby APs. These measurements are then compared with the previously collected fingerprint data in the database using an appropriate search or statistical matching algorithm. The generated result then becomes the location estimate for the user [6].

Note that this method does not require the development of an accurate RF signal propagation model. It avoids this by simply using the RSSI values directly, unlike the trilateration technique which converts the RSSI values to distances. The drawback of the fingerprinting based positioning technique is that it takes a significant amount of time to perform the offline data collection stage. Moreover, in dynamically changing indoor environments the movement of large furniture or machinery could negatively impact the positioning accuracy. To minimize such impacts the fingerprinting data may need to be updated as the indoor environment changes which could add to the cost of the system.

3) *k*-Nearest-Neighbor (KNN)

The *k*-Nearest-Neighbor (*KNN*) algorithm is a widely used technique for classification and other related applications. The main step in this algorithm involves the selection of the nearest k neighbors around the mobile user to determine its position [7]. For a Wi-Fi RSSI based positioning implementation, the *k*-nearest-neighbors are selected based on the RSSI values. After these values are sorted, the APs with the highest RSSI values are chosen as reference points for the position calculation. The value of k is typically selected experimentally by running multiple tests in the indoor environment. Other approaches, such as the one proposed by [7], modify the basic technique to enhance the *KNN* algorithm by dynamically adjusting the appropriate value of k.

Some implementations of *KNN* use weights to enhance the influence of close neighbors, these algorithms are called Weighted *K*-Nearest-Neighbor (*WKNN*). Because of its improved performance over the basic *KNN*, the *WKNN* technique is the one selected for prototyping the indoor positioning system at our university campus. The implementation details are provided in Section III.

C. *Other Technologies*

Even though BLE beacons and Wi-Fi are currently the most commonly used technologies for indoor positioning systems, researchers continue to explore alternative solutions. Such other options include geomagnetic [8], visual light communication (VLC) [9], ultra wideband (UWB) [10], and wearable sensors [11]. A more detailed examination and relative comparison of these techniques is provided in [1].

From the relative comparison of the different technologies, currently the Wi-Fi based solution offers one of the lowest deployment costs [1]. This is mainly because it can be developed using existing Wi-Fi infrastructure within a building to support wireless network access, thus there are no additional costs incurred to support the IPS. Similarly, unlike the other technologies, scaling the Wi-Fi based solution would not involve the deployment of additional infrastructure, if there are already existing access points in the environment. Moreover, since the APs are typically connected to the building's electrical power supply, there are no maintenance costs associated with replacement of batteries. One drawback of the Wi-Fi based solution is its limited accuracy, which on average is currently in the range of several meters.

A BLE beacon system with a high density of beacons could potentially offer a better accuracy than a Wi-Fi based solution. However, it would involve high deployment cost for the purchase of enough Bluetooth beacons to cover the entire indoor environment. At every point where determination of location information is desired, three or more beacons must be within the broadcasting range of the point. Due to their limited broadcasting range, BLE beacons are required every 7 to 10 meters, which means higher density of BLE beacons are required compared to more sparsely spaced Wi-Fi access points. Moreover, since most BLE beacons are battery powered, regular maintenance is needed which would add to the cost and complexity of the system. Thus, BLE based IPS is not as easily scalable as that based on existing Wi-Fi infrastructure.

III. WI-FI RSSI BASED INDOOR POSITIONING SYSTEM

As discussed in the previous section the Wi-Fi RSSI based technology was selected for the development and implementation of the indoor positioning system (IPS) at our University [12]. For the actual implementation, a weighted *k*-nearest-neighbor (*WKNN*) algorithm is developed using an Android app to generate the location estimate of the mobile user in real-time within a predesigned map.

A. WKNN Algorithm

The *WKNN* algorithm implemented in this prototype was inspired by Shin's enhanced weighted *k*-nearest-neighbor algorithm as presented in [7]. The pseudo-code for the algorithm implementation is given below:

1. Read RSSI values from the APs
2. Find the *k* APs with highest RSSI values
3. Use the smallest of the *k* RSSI values found in step 2 as the minimum threshold
4. Calculate the weights for each of the remaining (*k*-1) RSSI values using the following formula:

   $W_n = RSSI_n - Minimum\_threshold$

5. Multiply each AP's (X, Y) location by their calculated weight:

   $Weighted\_X_n = W_n * X_n$

   $Weighted\_Y_n = W_n * Y_n$

6. Calculate the total weights using Equation (4)
7. Calculate the *(x, y)* position estimate of the user by dividing the sum of the weighted coordinate values by the total of the weights, using Equation (5).

To demonstrate the operation of the *WKNN* algorithm, let's consider an example given in Figure 1. There are 5 access points (shown in blue) in the environment. The RSSI values that the mobile user (shown in red) reads from the 5 APs are given. Table 1 displays the (x_n, y_n) coordinates of each AP and the calculated values from the algorithm, for the total weight and the estimated *(x, y)* location of the user.

$$W_t = \sum_{n=1}^{k-1} W_n \tag{4}$$

$$(x, y) = \frac{\sum_{n=1}^{k-1}[W_n * (x_n, y_n)]}{W_t} \tag{5}$$

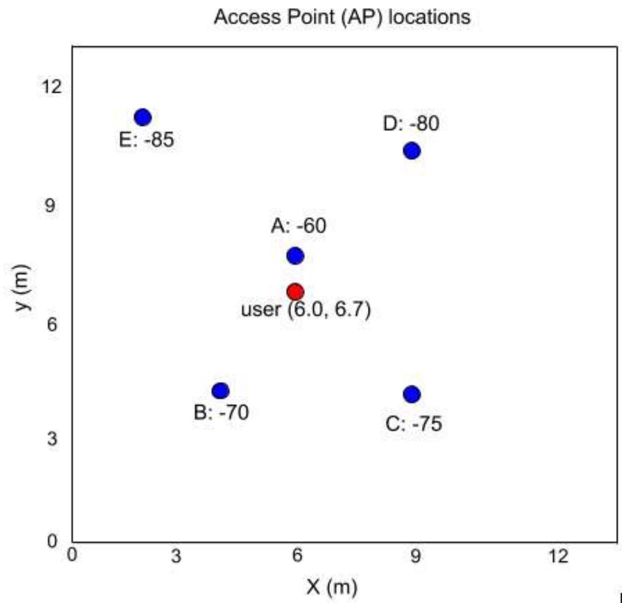

Figure 1. WKNN Example.

Table 1. *WKNN* Algorithm Example Data

AP	X (m)	Y (m)	Weight	Weighted X	Weighted Y
A	6.0	7.8	25	150	195
B	4.2	4.2	15	63	63
C	9.0	4.2	10	90	42
D	9.0	10.2	5	45	51
		Totals	55	438	351
			Final	6.33	6.38

In this example, the true position of the user is (6.0, 6.7). For the sake of simplicity of the example the value of *k* is set to 5, with the five APs shown within the surroundings of the user's location. The user's mobile device performs RSSI scan from all the APs within its close proximity to obtain the values shown in Figure 1. Note that the strength of the RSSI depends on distance and the transmission power of the radio signal source. Higher negative values correspond to relatively lower actual received power levels. Access point E has the lowest RSSI value at -85, thus this becomes the minimum threshold value, and for the rest of the calculations this AP won't be used any further.

Next, the weights for the access points A through D are calculated as described here. For example, for access point A, evaluate $RSSI_A$ - $RSSI_E$ = (-60) – (-85), which results in a weight value of 25. This process is repeated for all the APs. The next step in the *WKNN* algorithm is to take each weight and calculate the weighted X and weighted Y values. For example, for access point A, the weighed X value becomes 6.0 * 25 or 150. The next step is add all the weighted X's together and all the weighted Y's together, which are then divided by the sum of the weights to obtain the estimated location of the user.

The calculated position of the user generated by the *WKNN* algorithm is (6.33, 6.38), while the actual location of the user was (6.0, 6.7). The result shows a discrepancy from the true location of the user. The main contributing factor for not achieving high accuracy is because the *WKNN* algorithm does not take into account the very complex nature of the propagation of radio signals in indoor environments cluttered with objects and other electromagnetic waves that could potentially interfere with the Wi-Fi signals. The main reason for using this less accurate method is primarily to reduce the implementation cost and complexity.

In the experiments conducted it was learnt that the value of *k* in the *WKNN* algorithm affects the quality of the result. To determine an optimal value for *k*, a test experiment was conducted to calculate a stationary user's position for varying values of *k*. For this exercise, the actual position of the user was first set at a specific known location. The *WKNN* algorithm was then run for varying values of *k* with the location calculated for each case. It was then observed from the results of the experiment that the *k* value of 7 produces the most accurate position for the test compared to other values of *k*. Therefore, for the rest of the project implementation *k* was set to 7 for the *WKNN* algorithm based IPS prototype development.

The value of *k* in the *WKNN* implementation used for this project was determined ahead of time and then held fixed. In alternative approaches, the *k* value could be determined at run time. For example, in the enhanced weighted *k*-nearest-neighbor algorithm [7], the value of *k* is dynamically determined based on a threshold value for eliminating low RSSI signals from consideration for the position estimates.

B. Application Components

The mobile App for the indoor positioning system is developed using Android Studio. The screenshot of the Android app is shown in Figure 2. This section briefly explains the main components of the app.

1) Location indicator

A location indicator on the app is created to show the user's current location, which gets updated dynamically as the user moves around. Due to the possible errors in the location estimate as explained in the previous section, the user's location is represented in the map with a single blue dot surrounded by a

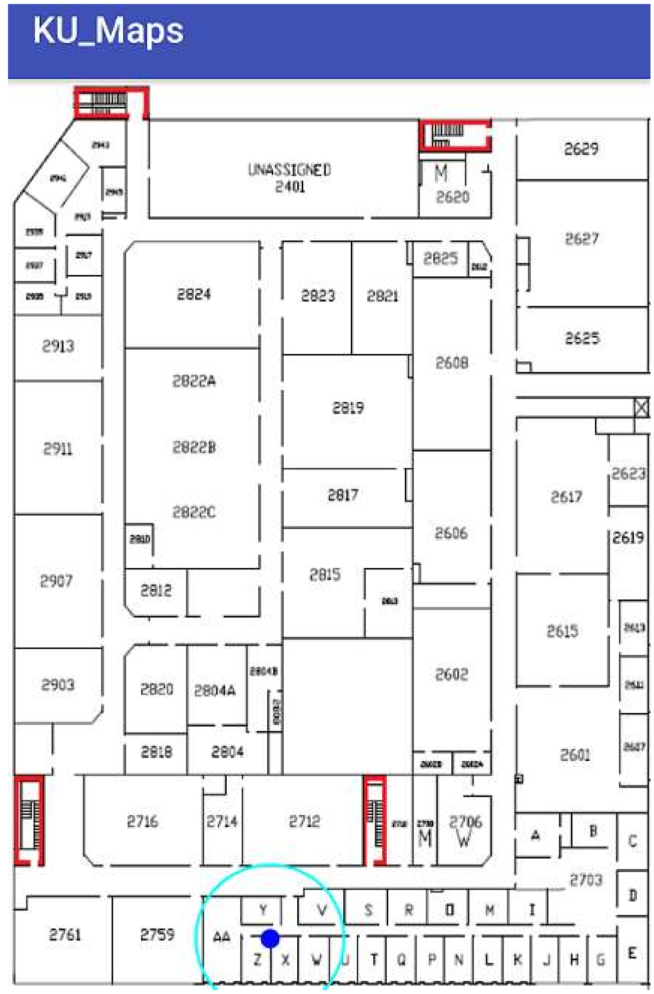

Figure 2. The KU_Maps app, the user is located within the cyan circle.

circle. The size of the circle corresponds to the uncertainty in the position estimate. The mobile application uses information about the display screen size of the smart device to maintain an appropriate and scaled size across different devices and pixel resolutions.

2) Zoom/Scroll

For intuitive user experience, the mobile app offers capabilities for zooming and scrolling features. The zoom scale factor and the transformation matrix of the view are available from the Android APIs. This information coupled with the scaling matrix are used for implementing a transformation function for translating a position in absolute (screen) coordinates to the overall map coordinates.

IV. PATH PLANNING

To support navigational services for mobile users it is important to implement a path planning algorithm. Planning a path for navigation can often be modeled as a search problem in a graph domain, for which a number of graph-based search algorithms have been created for obtaining the least-cost path in the graph with some defined cost parameters. In general, the most common path planning techniques can be categorized into deterministic, heuristic-based, and randomized algorithms [13]. A-star (A*) algorithm, which is one of the most popular graph-based search techniques for motion planning, was chosen for implementation in this project. A* is founded on the classic Dijkstra's graph search algorithm, with some heuristic- based optimizations applied to improve the performance of the search.

The result generated by the A* path planning algorithm provides an optimal route for a user navigating from their current location to a destination point. The development of the A* algorithm requires breaking up the two-dimensional environment into a map made up of small 2-D grids. The 2-D grids are then represented as graphs, with each block represented as a node in the graph and the connections between neighboring blocks become the edges. If a block contains a wall that block is not navigable and it is excluded from the graph search.

The A* algorithm generally provides the lowest cost path provided that the heuristics for estimating cost measures is defined properly. To simplify the A* implementation, a cost value of 1 was used for each horizontal and vertical movements, and diagonal movements were excluded. The A* algorithm terminates once the destination is found. What makes A* more efficient compared to other similar path planning algorithms is that it does not need to explore and evaluate every possible solution before returning the optimal solution, under the specified heuristics.

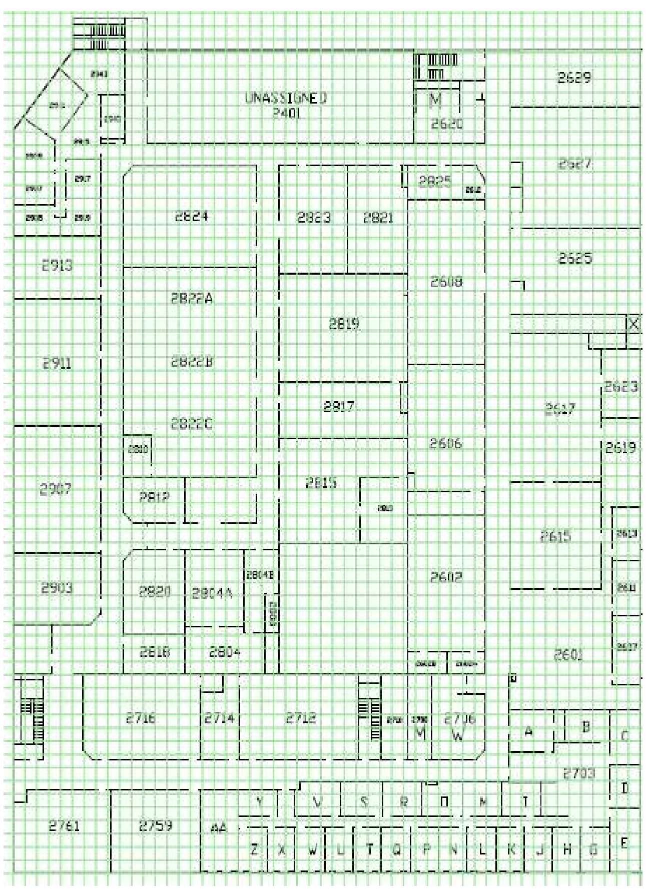

Figure 3. Grid of ECE Department Map

Figure 3 shows the grid representation for the environment under consideration which uses 3,564 blocks, 54 along X and 66 along Y. The map dimension is 65.8 m along X and 80.5 m along Y. Each block is 1.22 m by 1.22 m. This grid provides a resolution that fits one to two blocks in the width of a hallway, which allows for reasonable drawing of the generated path.

A higher resolution map could be used but it would be at the cost of increasing the computational time for the A* algorithm. For this prototype, the chosen resolution performs adequately. The computation is quick enough to not be noticeable by the end user while the resolution is fine enough to provide a path to any designated room on the map. The specifics of the A* algorithm will not be further explained in this paper. However, references [13] and [14] provide good explanations that help understand the basic theory and implementation ideas for the algorithm.

The A* algorithm implemented in this work has been observed to correctly return the least-cost path. Figure 4 demonstrates an illustration of the least-cost path generated by the algorithm from the user's currents position designated by the blue dot inside the cyan circle to the destination room (room # 2818).

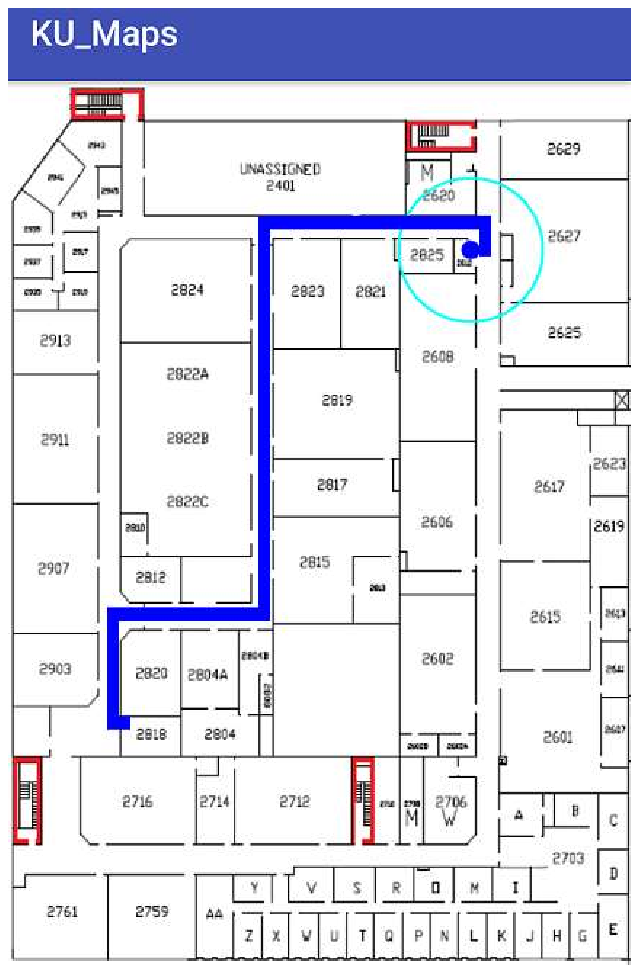

Figure 4. Path plan generated by the A* algorithm.

V. ANALYSIS AND DISCUSSION

The original goals of this indoor navigation project were to meet the following requirements:

1. Create mobile app with intuitive user interface
2. Map representation with zooming & scrolling
3. Show user position with acceptable accuracy
4. Path planning & routing capabilities
4. Scalable
6. Low cost

After the indoor navigation system prototype has been implemented, several experiments were conducted to evaluate its performance. The first three goals could be easily verified to have been successfully met by operating the app with proper user interactions. The app's GUI are as shown in Figure 2 and Figure 4.

To evaluate the localization accuracy, the user walks around to different locations in the environment while observing the calculated positions. These observed positions are then compared against the actual position of the user on the ground. After several experiments at different locations in the environment the average error of the position calculation is found to fall into two categories:

a) Average localization error in the range of 0 to 5 meters. These relatively good localization accuracy correspond to areas with good Wi-Fi coverage by multiple access points, and with little or no obstructions such as walls or other large equipment in the path of the radio signals. Parts of the hallways and areas near some of the classrooms have localization accuracies within this range.

b) Average localization error in the range of 5 to 10 meters. Such relatively lower accuracy correspond to areas that do not receive strong Wi-Fi signals from multiple access points. Also some areas face significant signal degradation from the concrete walls in the building and the heavy machinery inside some of the laboratories.

For improving the localization accuracy it would be possible to apply a different technique for Wi-Fi based localization, such as the fingerprinting approach. This could potentially help keep the error to less than 5 meters. However, the improved accuracy of this alternative method comes at a significantly higher cost in terms of major initial setup task which is needed to collect the fingerprinting data of the environment. Moreover, the fingerprinting data needs to be updated if the environment changes over time.

Another alternative option is to use a combination of multiple technologies that can help improve the accuracy and reliability of the system. For example, to provide additional reference information for the positioning algorithm, Bluetooth Low Energy (BLE) beacons could be used in the areas that have relatively higher inaccuracies due to Wi-Fi weak-spots. However, this method would incur additional cost to deploy the BLE beacons.

For further improvement of the position accuracy the mobile device's built in sensors could be utilized for dead-reckoning. Real-time data from sensors such as accelerometers and gyroscopes can be combined using sensor fusion algorithms to track the device's orientation and movement speed. If the direction and number of steps a user is taking from a reference point is calculated, this information could be used to improve the estimate for a user's location.

The prototype system presented in this paper is easily scalable to cover larger areas, by incorporating, in the design, the information about the Wi-Fi access points in the corresponding areas. It is also one of the most cost-effective technologies compared to the other ones discussed in Section III. The only cost of the system is the human capital needed to design and implement the system; assuming the environment already has existing Wi-Fi infrastructure, which could be assumed to be the case in most indoor environments these days.

Google Indoor Maps [15] is also found to be a promising tool that provides an API and development system for indoor positioning and navigation. This could potentially be a great resource for developing an INS prototype in the future. The API utilizes Wi-Fi infrastructure for the positioning. The advantage of this tool is that it makes it possible to access Google services such as routing and location estimation, which would not need to be created by the developer from scratch. It would also help reduce the complexity of the app development to handle features such as zooming and scrolling. Unfortunately, at the time or writing this paper, Google was mainly focused on supporting developments for large shopping malls, sports complexes, and airports; not small projects like that of ours.

VI. CONCLUSIONS

This paper presented a low-cost and effective Wi-Fi based indoor positioning system and utilized the A* algorithm for path planning. The localization method employed in the project relies on the availability of good Wi-Fi infrastructure in the indoor environment. The WKNN algorithm deployed for this purpose is found to be a relatively simpler approach compared to the alternative methods, such as the fingerprinting technique. Through the project prototype it has been demonstrated that the system has acceptable level of accuracy for many common indoor navigational applications.

A typical application scenario that is a target use case for this project is providing navigational aid for university campus visitors. Currently, our university trains and employs students to serve as tour guides for visiting prospective students and their parents. Campus visits are offered on a daily basis and they are often conducted by the tour guides to take the visitors to specific locations of their interest (labs, classrooms, offices, etc.) One of the main benefits of this INS app would be to offer visitors the opportunity to lead their own tour by freely exploring the campus at their own pace. In future work, the map coverage of the app could be expanded and additional multimedia material could be created to provide audio-visual guide for the visitors to enhance their campus visit experience.

REFERENCES

[1] J. Golenbiewski, G. Tewolde, and J. Kwon, "Evaluation of Indoor Positioning Technologies for Prototyping at Kettering University," in *The 2018 IEEE Internation Conference on Electro/Information Technology (EIT 2018)*.

[2] "iBeacon," [Online]. Available: https://developer.apple.com/ibeacon/.

[3] "Google Beacon Platform," [Online]. Available: https://developers.google.com/beacons/.

[4] "How to do accurate indoor positioning with Bluetooth beacons?," [Online]. Available: https://proximi.io/accurate-indoor-positioning-bluetooth-beacons/.

[5] O. Pathak, P. Palaskar, R. Palkar, and M. Tawari, "Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points --A Tri-lateration Approach," *International Journal of Scientific & Engineering Research,* vol. 5, no. 4, pp. 1234-1238, 2014.

[6] C. Rizos, A. G. Dempster, B. Li, and J. Salter, "Indoor positioning techniques based on wireless LAN," in *LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications,* 2007.

[7] B. Shin, J. H. Lee, T. Lee, and H. S. Kim, "Enhanced weighted K-nearest neighbor algorithm for indoor Wi-Fi positioning systems," in *The 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, 2012.

[8] IndoorAtlas, "IndoorAtlas," [Online]. Available: https://www.indooratlas.com/.

[9] "Indoor Positioning - Perferct Light, Precise Location," [Online]. Available: http://www.lighting.philips.com/main/systems/themes/led-based-indoor-positioning.

[10] Pozyx, "How does ultra-wideband work?," [Online]. Available: https://www.pozyx.io/pozyx-academy/how-does-uwb-work.

[11] A. Suprem, V. Deep and T. Elarabi, "Orientation and Displacement Detection for Smartphone Device Based IMUs," *IEEE Access,* vol. 5, pp. 987-997, 2017.

[12] J. Golenbiewski and G. Tewolde, "Implementation of an Indoor Positioning System using the WKNN Algorithm," in *The 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC 2019)*, 2019.

[13] D. Ferguson, M. Likhachev, and A. Stentz, "A Guide to Heuristic-based Path Planning," in *The International Conference on Automated Planning & Scheduling (ICAPS)*, 2005.

[14] A* algorithms. [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming/.

[15] "Go Inside with Indoor Maps," [Online]. Available: https://www.google.com/maps/about/partners/indoormaps/.