Multi-Device Sensor Fusion with Inertial Data and WLAN Signals for Improved Indoor Localization

Bhatt, Isha
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, MI
ibhatt@umich.edu

Minnehan, Gillian
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, MI
gminn@umich.edu

Radhakrishnan, Sudharshna
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, MI
sradha@umich.edu

Thomas, Sharon
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, MI
sharonrt@umich.edu

***Abstract*—Indoor localization of individuals is a challenging task requiring novel approaches to effectively balance cost, comfort, and accuracy. In this paper, we investigate position estimation techniques that would enable indoor localization of warehouse employees for increased safety. Inertial measurement units (IMUs) are commonly used in existing indoor positioning solutions, however they are notorious for accumulating error due to integration of acceleration measurements over time, therefore we investigate a more accurate velocity estimation using a deep convolutional neural network. For a further improvement in accuracy, we also explore multi-device data fusion using a smartphone and smartwatch by averaging each of the position estimations obtained by integrating IMU data into one final estimation. Many existing indoor positioning systems rely on complex and expensive external sensors such as cameras and laser scanners to provide a periodic ground truth update to further correct drift from IMUs over time. By developing external positioning system based on received signal strength indicator (RSSI) values, we demonstrate position accuracy that is within bounds of the warehouse use case, and acts as a viable, low-cost alternative to existing external positioning systems.**

***Index Terms*—indoor localization, WLAN, IMU, deep neural network**

I. Introduction

In 2019, 4.8% of warehouse workers were injured and 24 fatalities were reported in the United States [source]. There has also been an overall increase in fatalities from 2016. To improve warehouse safety, indoor tracking of individual workers is proposed in this study so that workers' locations are known during emergency situations and when moving large, potentially dangerous payloads.

Warehouse technology must leverage localization techniques tailored to indoor environments in order to enable employee tracking. Global Navigation Satellite Systems (GNSS) are incredibly successful and widely used to locate devices being held or worn by the person being located in outdoor environments, but indoors, GNSS signals are not receivable or usable. Indoor localization of individuals poses a number of challenges. Firstly, indoor environments are often crowded with many people and objects, which alter wireless signal propagation [2]. Secondly, applications where indoor localization has been proposed, such as elderly care, hospital patient tracking, and firefighter tracking, require solutions to be real-time, non-invasive, and highly accurate. Many existing solutions fail to meet most of these requirements.

Developing inertial navigation systems for indoor localization has been the subject of intense interest by researchers. Many positioning systems use a number of effective approaches such as infrared, ultrasonic, RFIT, Ultra Wideband, ZigBee, and computer vision [3], but none are suitable for the mobile devices that warehouse employees could carry or wear. As an alternative, inertial navigation can use data from accelerometers and gyroscopes to obtain a dead reckoning estimate of position and orientation through numerical integration of acceleration and angular velocity. Using the hardware in smartphones and smartwatches is less expensive than these more advanced sensor systems, and moreover, their ubiquity presents an opportunity to lower costs of positioning systems overall. The main disadvantage to this method is that integrating data using biased measurements results in drifts that hamper position and orientation estimations over time. To correct this drift, some solutions use external positioning systems to provide a correction to the position estimation every few seconds [2].

Another proposed method for indoor localization leverages wireless area networks (WLANs) to find device position. WLANs have been used because of their low additional cost to existing infrastructure. One major technique for RSSI-based localization is fingerprinting. In Wi-Fi fingerprinting, a signature is constructed through experiments to model the relationship between the user position and the strength of the wireless local area network (WLAN) signals received by the user. Alternatively, use of time of arrival measurements have shown promise as an improvement over the more variable

RSSI measurements. However, accuracy falls significantly with this technique in environments with severe non-line-of-sight [4].

In this paper, we investigate several modifications to the indoor positioning system presented in [2]. First, we explore the use of Wi-Fi RSSI fingerprinting to obtain a position estimation that can be used as a low-cost, simple alternative to the fixed camera monitoring system and Baysian estimation presented by Colombo et al. Secondly, we investigated an alternative velocity estimation scheme that averages the velocity estimated by passing IMU data from a smartphone and smartwatch separately into a Deep Convolutional Neural Network (DCNN) constructed for walking speed estimation [5]. We begin by examining related work in using WLAN signals and IMU Data for position estimation in Section II. In Section III, we describe our structure of our experiments including methods of data collection and our assumptions. In Section IV, we describe our methods and how they would be integrated into a complete indoor positioning system. In Section V, we analyze the results of our experiments. Finally, in Section VI we describe future work that can be done to explore the use of RSSI measurements, IMU data from multiple devices, and DCNNs for indoor positioning.

II. Related Work

There have been numerous papers on indoor localization with various devices that leverage inertial data, WLAN signals, or both. Some papers using inertial data also discuss methods for velocity estimation, which is relevant for accurate indoor localization. In this section we will describe works that are most similar to ours, particularly those we build off of.

A. Localization using WLAN Signals

Location of individuals using WLAN signals has become increasingly more common given the ubiquity of wireless networks in buildings. Many researchers have used received signal strength indicator (RSSI) measurements from multiple wireless network access points to gain a rough estimate of location [6] [7] [8]. RSSI indicates the power level received after any possible loss at the antenna and cable level. The closer the RSSI value is to 0, the stronger the signal [9]. Golestani et al. achieved position estimation accuracy of within three meters with RSSI measurements using a path loss exponent estimation algorithm [6]. While studies leveraging RSSI measurements such as Golestani et al. have been shown to produce adequate results for some applications, use of time of arrival (TOA) data shows immense promise in leveraging WLAN signals for positioning [10]. Use of TOA data in positioning systems has increased, but is limited by the availability of these measurements for mobile developers and the capability of the access points, which must be able to transmit TOA measurements. Google has made an application available for Android developers that gives TOA data, thus enabling widespread use of TOA measurements for positioning of mobile devices [11], however, Apple has yet to make this information available to iOS developers. Our work exclusively leverages Apple products, therefore we are limited to RSSI measurements only, but recent developer communities have hinted at the possibility of more localization data becoming available to developers in the near future [12]. As mentioned earlier, TOA measurements would not work well in warehouses since it is an extremely non-line-of-sight environment.

B. Localization using IMU Data

Internal measurement units (IMUs) contain multiple sensors including a gyroscope, accelerometer, and magnetometer. They are used to measure the distance covered by a moving object in a reference frame. However, positioning using only IMU data is known to be inaccurate over time due to unavoidable uncertainty growth [2] [13], therefore IMU-based localization solutions are often coupled with other sensors, some of which act as an external positioning (EP) system for periodic corrections of the drift from integration of IMU measurements. Colombo et al. developed a cascading EKF position estimation algorithm that finds position using IMU data on a device worn around the waist, but requires an external position update from a complex vision tracking system containing four cameras every one to five seconds to correct for drift [2]. The localization is accurate for several seconds, and is considered a reasonable tradeoff between accuracy, cost, and comfort. The proposed system in this paper leverages more ubiquitous devices – iPhones and Apple Watches – making it more suitable for a large scale deployment than a custom device worn around the waist. Ahmed et al. [13] leverages both IMU data and Wi-fi fingerprinting for localization of warehouse robots, but their system also requires the use of wheel encoders, which are not practical for localization of people.

Instead of using an external positioning to increase accuracy, Yuan et. al [3] leverages machine learning techniques to predict pedestrian location using co-occurrence IMU data from a smartwatch and smartphone. The novelty of improved localization via data fusion between a wearable device and a phone is similar to the system presented in this paper, however, experiments were only conducted on an LG watch and Android phone. We propose testing on Apple products in addition to leveraging Wifi access points for increased accuracy. With Apple watches becoming more prominent, it is prudent to test the combination of an iPhone and Apple watch. Moreover, Yuan et. al [3] did not test with specific application in mind, while our project focuses on the warehouse setting. To our knowledge no work has used RSSI measurements as an external positioning system in addition to multi-device sensor fusion.

C. Velocity Estimation using IMU Data and a DCNN

Existing indoor positioning system methods require accurate speed estimation from inertial data, specifically from accelerometer and gyroscope data. Ideally, accelerometer data can be mathematically integrated once to get an accurate estimate of the object's velocity. However, this method is proven insufficient to accurately estimate an object's velocity due to the drift accumulated over time when integrating inertial sensor measurements. Recent advances have enabled more accurate velocity estimation. Most solutions in the field are based on body-mounted motion sensors. However, smartphones and watches are more scalable for large-scale deployments since they are more ubiquitous. A recent study showed that advanced neural networks can be used for accurate speed estimation [5]. In existing methods that use neural networks for velocity estimation, the root-mean squared-error (RMSE) is comparable to RMSE values from existing body-mounted sensor speed estimation methods. Due to its non-invasive nature and overall accuracy, we investigated the DeepWalking system to estimate the user's velocity.

III. Experimental Setup

Due to the dependency between the methods and the experimental setup, details on data collection are presented before methods. For RSSI and IMU data collection, a 9 m by 6 m obstacle-free area was mapped out in a large indoor space. Four TP-Link N300 Wireless Portable Nano Travel Routers are placed on each individual corner of the area, with Node 1 at the coordinate (0, 0), Node 2 at (9, 0), Node 3 at (9, 6), and Node 4 at (0, 6). The iPhone was used to take multiple readings of the RSSI strength at each 0.5 meter increment from each node using AirportUtility, an iOS application that finds RSSI values from all reachable Wi-Fi access points. The readings for each node at each increment were averaged and recorded in a reference database.

Three paths were chosen for testing, as seen in Figure 1. During data collection, the Apple Watch was laid on top of the iPhone and both were held flat as we walked through each path to ensure that devices were oriented the same and subject to the same movements. Both SensorLog, an iOS application used to record data from the IMU sensors in Apple devices, and AirportUtility were running simultaneously on the devices to collect data for the RSSI external positioning system and for velocity estimation.

Seven minutes of IMU training data was collected on a random path within the data collection area and three minutes of testing data was collected on the three predetermined paths from Figure 1. While the DeepWalking system uses a treadmill for the training and testing velocity data, we took time and distance measurements and calculated velocity while the user was on the path since the ultimate goal is to estimate the user's position. We did not use a treadmill for the training data to ensure consistency when training the model for predictions. Instead, the user walks on a given path in the grid for training or testing. The time it takes the user to walk between each coordinate point is recorded. The walking speed is then calculated by dividing the distance traveled between each coordinate point by the duration of time the user spent in that space between the coordinate points. For each diagonal segment that the user walks in the area, a distance of $\sqrt{2}$ meter is used. For each segment the user walks in the area that is parallel to the vertical or horizontal axes, a distance of 1 meter is used.

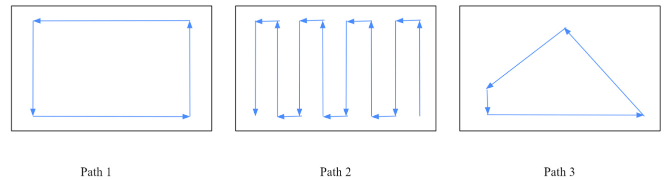

Fig. 1: RSSI and velocity estimation testing paths

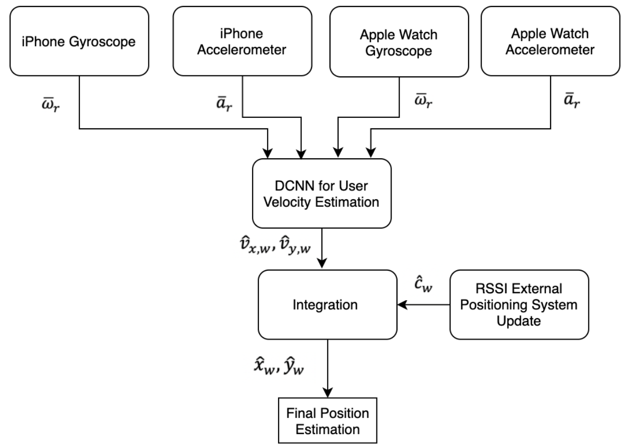

Fig. 2: Indoor positioning estimation block diagram.

IV. Methods

The proposed indoor localization system involves two primary methods: velocity estimation using IMU data fed into a deep neural network and position estimation using RSSI fingerprinting, which serves as a ground truth external positioning system. A block diagram of the complete proposed indoor localization system is shown in Figure 2. The velocity estimation method leverages a deep convolutional neural network (DCNN) to produce a walking speed estimate from the iPhone and Apple Watch separately, then averages the two to obtain a more accurate estimation. Then, the result of integrating the x and y velocity components is final position estimation. RSSI measurements from multiple Wi-Fi access points determine a location estimate that is treated as near-ground truth. When a new RSSI measurement is available, the system will recalibrate to this new measurement, therefore correcting for drift that has accumulated through repeated integrations of velocity.

Let $< W >$ be a global or world reference frame with values in this frame denoted with a w subscript (e.g. x_w), and $< R >$ be the frame attached to the inertial platform containing a gyroscope and accelerometer. For the sake of clarity, all symbols topped with a bar represent the measurements collected from the sensors (e.g. $\bar{w_r}$), while hatted symbols represent the values estimated by an algorithm (e.g. $\hat{v}_x$). In the rest of this section, we will elaborate on the velocity estimation using IMU data and a DCNN, and position estimation using RSSI fingerprinting.

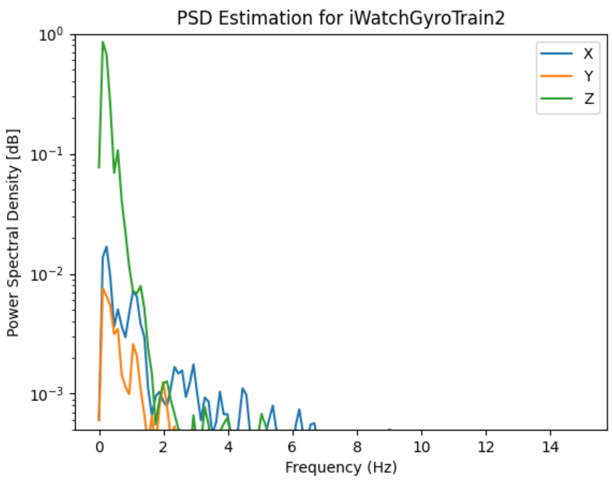

Fig. 3: Power Spectral Density Estimation for Apple Watch gyroscope from second minute of training data for all three axes.

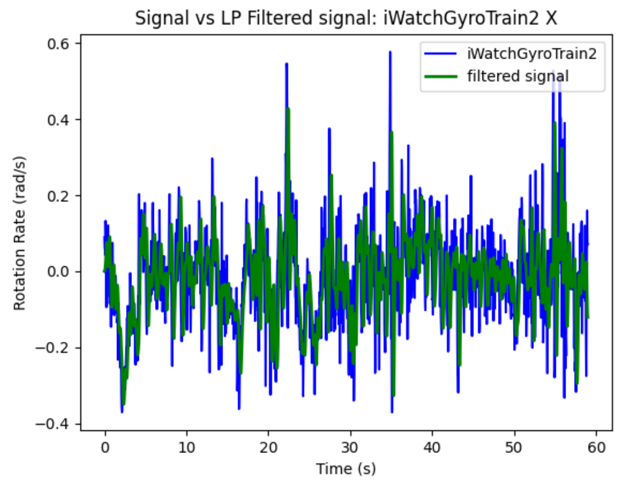

Fig. 4: Raw Apple Watch gyroscope data for X axis plotted against filtered data with cutoff frequency of 3 Hz.

A. *Velocity Estimation*

The proposed velocity estimation method leverages a DCNN that is trained offline and makes predictions offline. It is largely modeled after the DeepWalking system with several modifications for this application [5]. Similarly to the original DeepWalking system, the proposed velocity estimation system consists of five key components: data collection, filtering, coordinate system alignment, DCNN training, and DCNN prediction.

There are several notable differences between data collection methods for this paper and those proposed by Shrestha et. al [5]. During setup, data is collected on an iPhone XR instead of the Samsung Galaxy S6, but the DeepWalking system was largely unchanged since it can be easily ported to any mobile platform including iOS. The raw accelerometer and gyroscope data is collected for both the Apple Watch and the iPhone using the SensorLog application. Unix time for each sample was also collected for the iPhone. An Apple Watch Series 3 was used, which only has an accelerometer and gyroscope, so data on all three axes for the accelerometer and gyroscope are collected and the time for each sample is converted to Unix time. All inertial data is collected with a sampling frequency of 30 Hz. Another difference is that the original DeepWalking system architecture consists of a network model that is trained offline and predictions of walking speed are performed online. Predictions for this study are done offline, however future work would make this prediction phase online.

All five components of the DeepWalking algorithm are implemented in Python using the NumPy and Keras libraries. The filtering method used is identical to [5]. Using Welch's method with a Hanning window of 1 second and a half window overlap, the power spectral density is plotted for all three axes for both the accelerometers and gyroscopes for iPhone and Apple Watch. An example of this is shown in Figure 3.

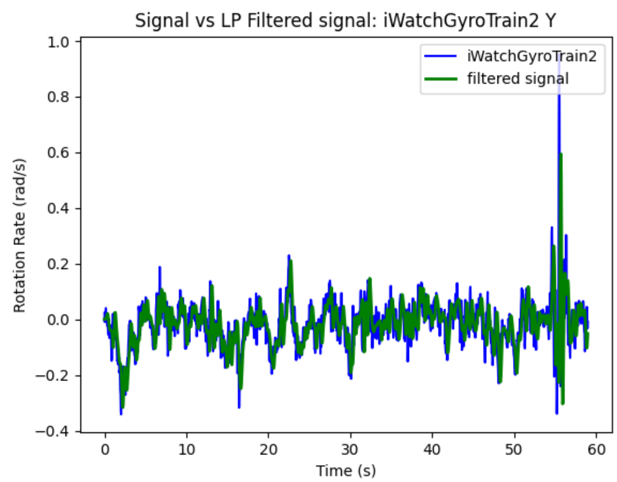

Fig. 5: Raw Apple Watch gyroscope data for Y axis plotted against filtered data with cutoff frequency of 3 Hz.

A cutoff frequency is determined and the sensor data is filtered for each axis for the given device and sensor. The filtered data for each axis for the Apple Watch gyroscope is shown in Figures 4-6.

The gyroscope and accelerometer data are then expressed on a fixed coordinate system through coordinate system alignment. The coordinate system alignment method is taken directly from the algorithm and left unchanged for our implementation. The gravity component is first derived by taking averages of measurements on each axis using N samples.

$$v_x = \frac{\sum_{i=0}^{N} a_x}{N} v_y = \frac{\sum_{i=0}^{N} a_y}{N} v_z = \frac{\sum_{i=0}^{N} a_z}{N}$$

The dynamic component of the averages is then found by subtracting the filtered sensor values from the gravity component

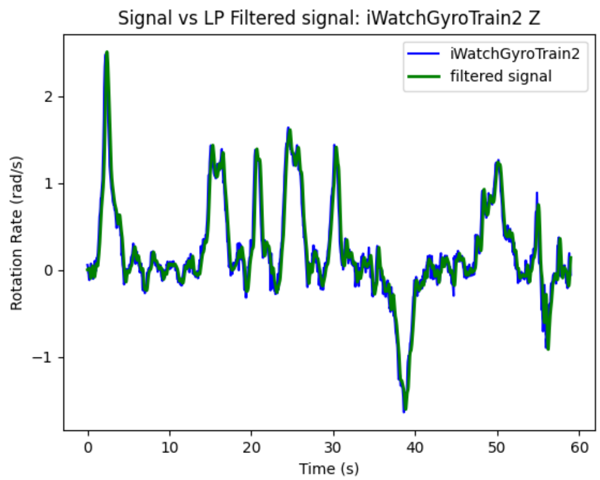

Fig. 6: Raw Apple Watch gyroscope data for Z axis plotted against filtered data with cutoff frequency of 3 Hz.

for all three axes.

$$d = \{a_x, a_y, a_z\}$$

After the dynamic component is found, the vertical component is calculated by projecting d onto the vertical axis v.

$$p = \frac{d \cdot v}{v \cdot v} v$$

The horizontal component is then calculated using the dynamic component and projection.

$$h = d - p$$

The procedure for coordinate system alignment is repeated for the gyroscope for similar system alignment. Once the horizontal and vertical components for the devices and sensors are found, the magnitude of each component is found and fed into the DCNN as images. The input to the DCNN is the number of N samples with 2x2x1 images where each image consists of four values that represents the magnitudes of the horizontal and vertical components of the accelerometer and gyroscope [5]. The neural network then consists of four cascaded groups of layers. Each group of layers consists of convolutional, batch normalization, ReLU, and Max Pooling layers followed by dropout, fully-connected and regression layers. In each cascaded group, there are 16, 32, 48 and 64 convolutional filters respectively. The stride used is 1 and the kernel size used is (2, 2). The dropout rate of 0.2 is kept the same. RMSE is also kept as the loss function [5]. The DCNN is applied on the raw data separately for the Apple Watch and iPhone for a velocity estimation, and the estimations are averaged and fed into the rest of the system.

B. External Position Estimation using RSSI Fingerprinting

The concept of creating a database of signal strength information was derived from Kothari et al. [14] in their paper on robust indoor localization on a commercial smart phone. Although this process is tedious, it only needs to be done once after installation of Wi-Fi range extenders. Kothari et al. [14] utilized a Euclidean distance metric to estimate position, and calculated a weighted average to reduce noise.

The position corresponding to the vector from the database with the smallest Euclidean distance to the measured RSSI vector is considered a possible position estimation. We have developed an algorithm in MATLAB that will return an averaged coordinate value, which will be used along with the IMU data to estimate the current position. The MATLAB algorithm takes four 12x18 matrices, one for each node, and calculates the range of acceptable values for the inputted RSSI value depending on the threshold. Then, it performs a logical AND with all four matrices to form a matrix where an element corresponds to a potential coordinate. These coordinates are translated to corresponding coordinates on the Cartesian grid, where it is $\frac{x-1}{2}, \frac{y-1}{2}$ of the original coordinates. This formula is used to correct for the corresponding coordinate because matrices are indexed starting at 1 in MATLAB, and the values in the matrix correspond to every 0.5 m in the experimental grid. The average of all potential coordinates is outputted as the final position estimation coordinate.

V. Evaluation and Results

A. RSSI Fingerprinting Results

As shown in Figures 7-9, we tested the accuracy of the RSSI fingerprinting method on three different paths. The Euclidean distance error between the RSSI estimated position and true position was calculated and plotted as shown in Figure 9. Each graph shows all three paths, with each colored bar corresponding to the average error given a particular threshold of time from the actual arrival at a known location on the path that an RSSI measurement is accepted. This threshold is needed given the infrequent RSSI measurement update. For example, the 0.5 threshold represents the error between RSSI values collected within 0.5 seconds of the user arriving at the known coordinate along the path and the known location. For the x-coordinate, the errors are 1.18 m, 1.15 m, and 1.35 m for Path 1, Path 2, and Path 3 respectively. For the y-coordinate, the errors are 2.46 m, 1.09 m, and 1.64 m for Path 1, Path 2 and Path 3, respectively. The average error for Euclidean distance is 2.74 m, 1.59 m, and 2.16 m for Path 1, Path 2 and Path 3 respectively. Thus, average error across all paths is around 1.5 m, with Path 2 having the least error. This error is reasonable for positioning in large indoor areas and for use as a ground truth position in an external positioning system Intuitively, there is also a very small correlation between time threshold and error, indicating that there is less error when the threshold is smaller.

B. Multi-Device Velocity Estimation

When the velocity estimation model was evaluated, we found that the model was very inaccurate. For the iPhone, the model's accuracy was 9.82% and for the Apple Watch, the model's accuracy was 15.78%. Plots for actual and predicted velocity are shown in Figures 10-15.

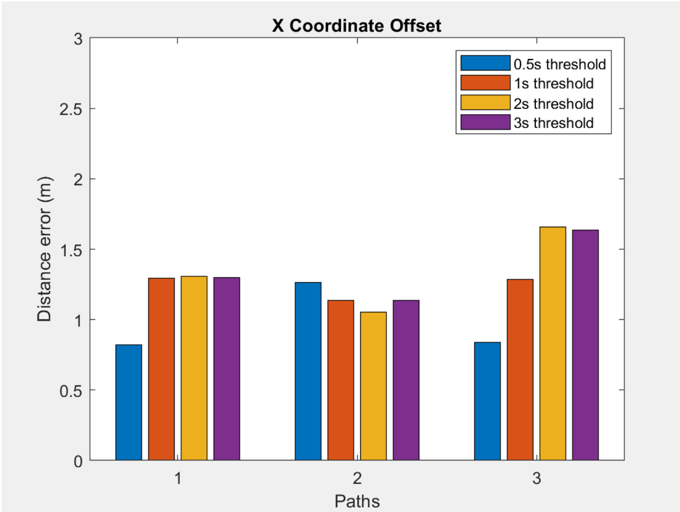

Fig. 7: X-coordinate average offset in meters.

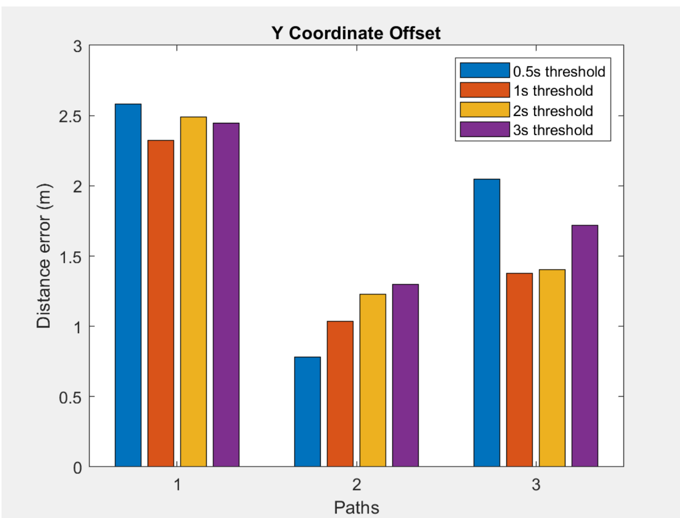

Fig. 8: Y-coordinate average offset in meters.

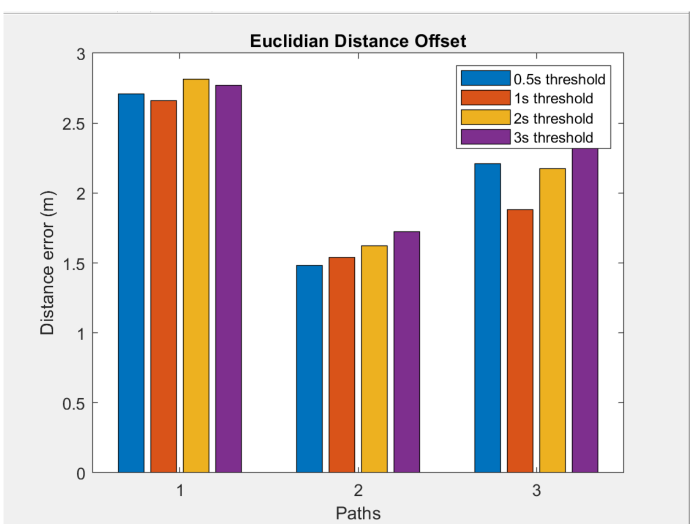

Fig. 9: Euclidean distance average offset in meters.

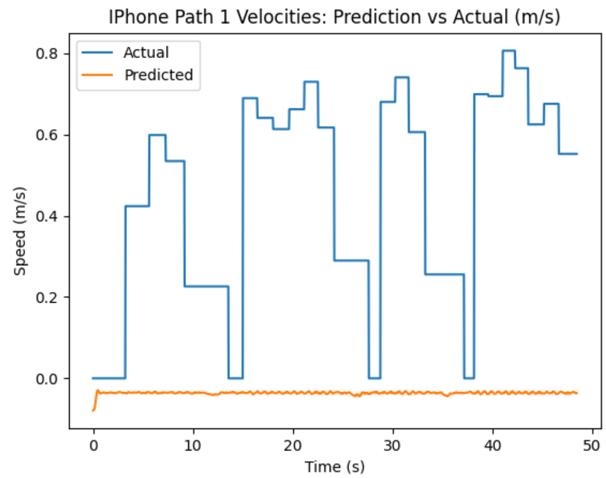

Fig. 10: Path 1 velocity prediction versus actual for iPhone.

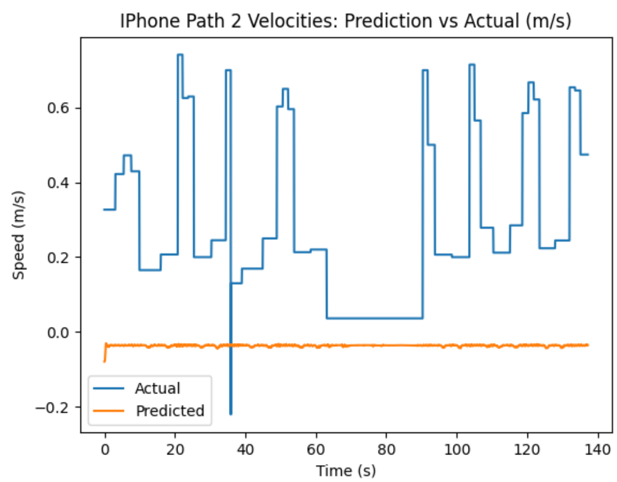

Fig. 11: Path 2 velocity prediction versus actual for iPhone.

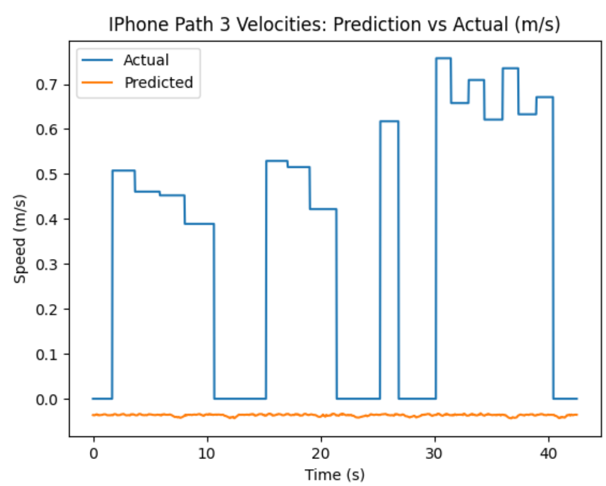

Fig. 12: Path 3 velocity prediction versus actual for iPhone.

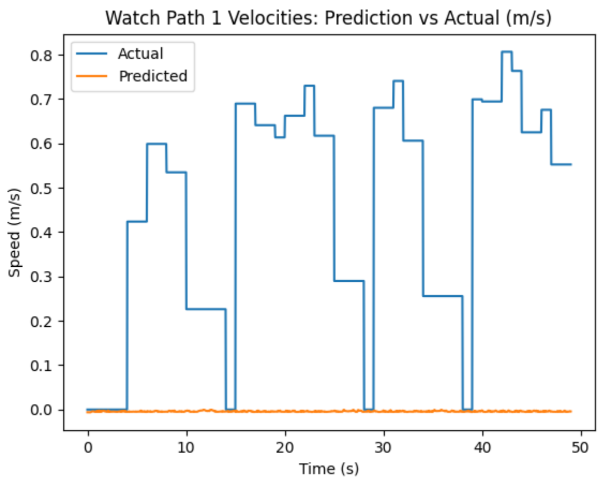

Fig. 13: Path 1 velocity prediction versus actual for Apple Watch.

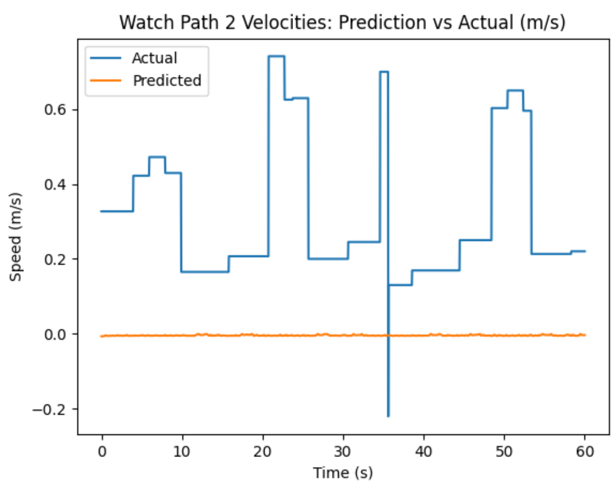

Fig. 14: Path 2 Velocity Prediction versus Actual for Apple Watch.

With more development time and future research, the model can be tuned by testing out different image shape configurations. In the DeepWalking system, they tried different configurations for M, therefore we can rule out the difference in the number of images used in our system versus theirs as a source of prediction error. However it is unclear why in the original DeepWalking system an input image size of 45x4x1 is chosen for M images. This detail may be a source of error when porting their solution to our application. It may also be possible that our velocity data collection method is not compatible with the existing DeepWalking model since their treadmill velocity data was constant for any given data trace collection while ours was variable. Ideally, a future iteration would be to tune the neural network to estimate velocity with high accuracy, therefore allowing the integration

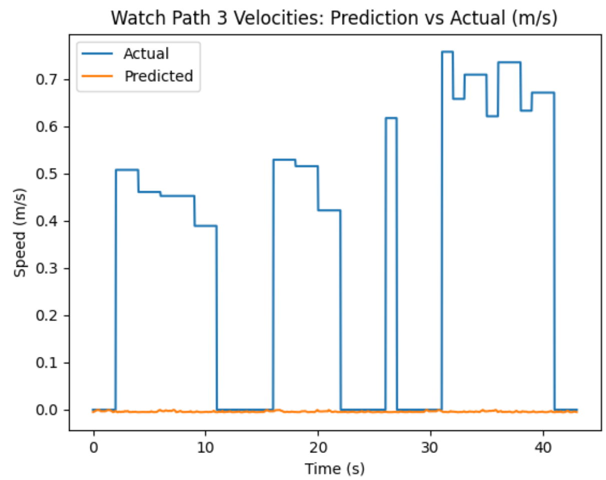

Fig. 15: Path 3 Velocity Prediction versus Actual for Apple Watch.

of the estimation once to get position, and the correction of the estimation every 3 seconds with the external positioning system.

C. Integrated RSSI and Multi-Device Position Estimation

Due to the low accuracy of the velocity estimation model, the discussed velocity estimation was not used to evaluate the performance of multi-device sensor fusion and RSSI external positioning system. The position estimation using IMU data was instead obtained by first rotating then double integrating measurements from each device's accelerometer to obtain separate position estimations. When an RSSI position update was available, it was applied to only the iPhone position estimation since the Apple Watch did not have an application available for obtaining RSSI values. A block diagram of the system used to obtain the results in this section is seen in Figure 16, which is a modification of the original block diagram from Figure 16.

To obtain the acceleration vector in the world frame, the standard rotation matrix was applied using the roll, pitch, and yaw values found by integrating the measured angular velocity values from the gyroscope. Once rotated, the acceleration vector was integrated twice over the time duration between the initial measurement and the measurement following the current measurement being considered. Since this positioning system is only concerned with the xy-plane, the z-axis data is ignored. The result of double integration was the displacement in x and in y for that time frame; the initial x and y positions are added to these values, respectively, to obtain the position estimation. RSSI measurements were available about every three to four seconds. They were accepted and applied to the iPhone position estimation if they were collected within a certain threshold of the time the iPhone IMU data was collected. Intuitively, as the threshold increased, the number of RSSI measurements that were accepted also increased. To evaluate the effect of an

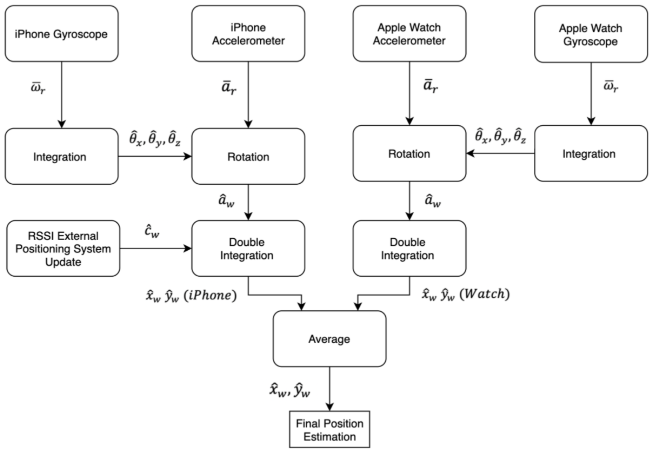

Fig. 16: Indoor position estimation block diagram modified for testing.

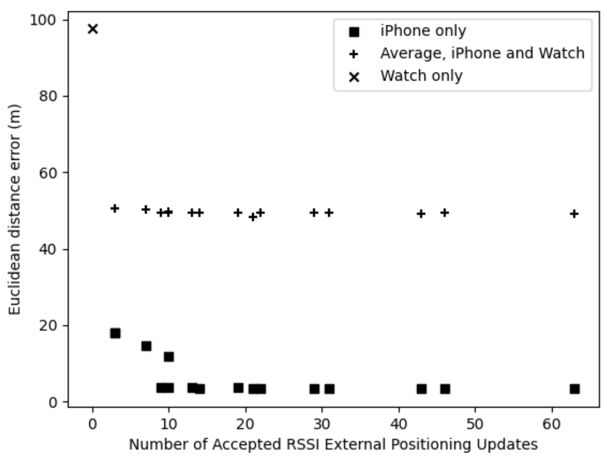

Fig. 17: Errors for Path 1 (perimeter).

increased number of RSSI measurements on accuracy, as well as the accuracy achieved when the iPhone and Apple Watch were averaged, average position estimation accuracy versus the number of RSSI measurements for each path were computed and can be seen in Figures 17, 18, and 19. Note that there is only one point for the watch since RSSI values for the watch were not collected. Future work could involve obtaining RSSI values for the watch so that each device leverages WLAN signals. Each point represents a different threshold.

For Path 1 and 2, there is a notable decrease in error as the number of RSSI positions applied increases for the iPhone. This trend is most likely due to the significant error caused by drift, and the fact that applying RSSI position estimations to the displacement calculated by integration of IMU data for up to three seconds after a said RSSI position estimation was collected increases accuracy. Error is relatively constant as the number of RSSI positions applied increases for Path

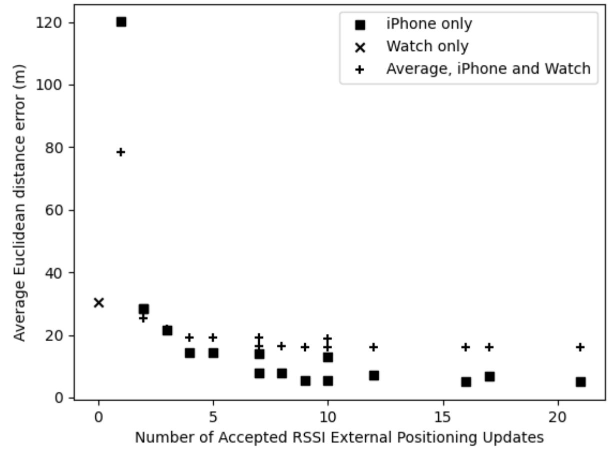

Fig. 18: Errors for Path 2 (zigzag).

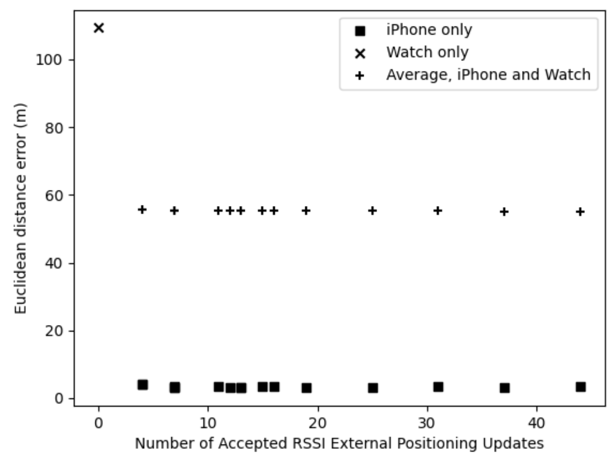

Fig. 19: Errors for Path 3 (angle).

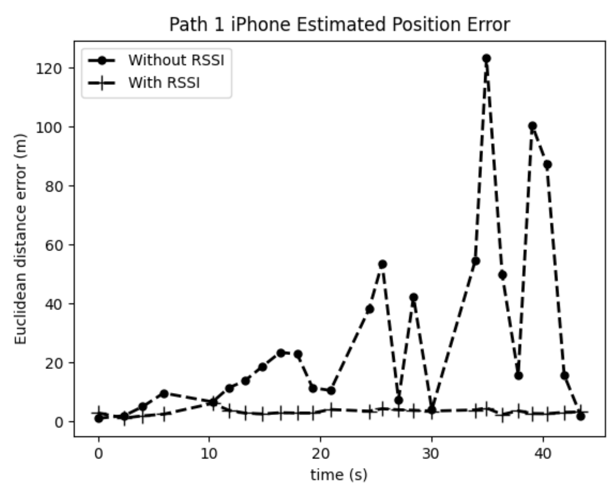

Fig. 20: Comparison of errors in Path 1 with and without RSSI.

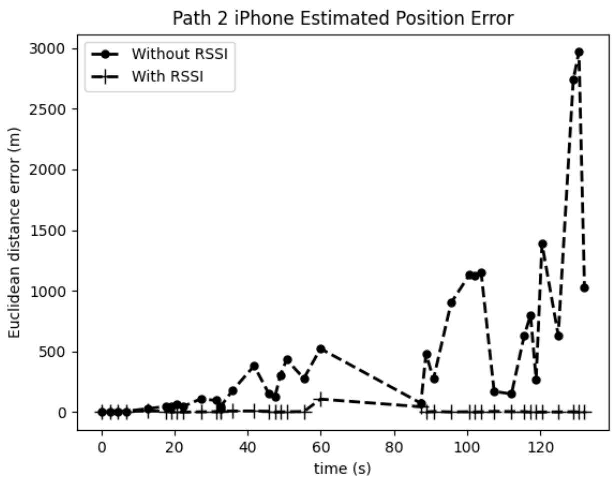

Fig. 21: Comparison of errors in Path 2 with and without RSSI.

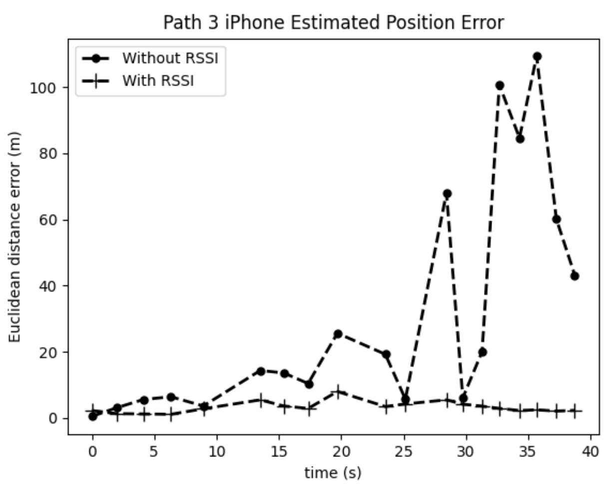

Fig. 22: Comparison of errors in Path 3 with and without RSSI.

3, which is expected since the initial position estimation was fairly accurate. However, it is not clear why the initial position estimate was highly accurate only Path 3. For all paths, the iPhone position estimations were generally more accurate, with the exception of the initial position estimation for the iPhone in Path 2 where only one RSSI position update was accepted. Without a way to correct the drift of the watch data over time, a high average error from the watch across the whole path is expected. It could be hypothesized that obtaining RSSI values from the watch would reduce error similarly to what was observed for the iPhone. For Path 1 and Path 3, averaging the watch and iPhone data produced an estimation that was much worse than the position estimation for the watch alone. However, for Path 2, when the number of RSSI position estimations applied was low (less than four), the average was better than or equal to that of the iPhone since the watch was more accurate to start. These results indicate the potential advantage in averaging the two when it is unknown which will be inaccurate at a given time. Further work to investigate this hypothesis could include first obtaining RSSI values for the watch to place the iPhone and watch on the same playing field as far as accuracy, and then conducting more path tests to evaluate the result of averaging iPhone and watch position estimations.

To further illustrate the estimation improvement from RSSI, the position estimation over time for each path when RSSI is used and is not used is shown in Figures 20, 21, 22. These paths were evaluated at the threshold that produced the smallest average error.

VI. Future Work

A. Elimination of Assumptions

This study produced a system that was not real-time. However, localization in warehouses and for other applications is expected to be real-time. Future work to fully evaluate the viability of this system for the given use case would be to make the system real-time.

B. Further RSSI Hypotheses

While the measurements taken during this experiment were relatively accurate, the accuracy of the system could be increased with measurements taken at smaller distance increments. The system could also be tested with and without people or other obstacles in the controlled area to see how that affects the system's accuracy.

C. Kalman Filter

Kalman filters are often used to smooth out noisy measurements for more accurate estimation of state [Colombo]. Many papers that used IMU data for position estimation leveraged Kalman filters or Extended Kalman filters (EKF) [2][13]. Originally, this project proposed implementing the cascaded EKF system from [2], however the full implementation of these filters, which requires careful tuning of covariance matrices modeling trust in sensor data, is left to future research. A block diagram of the proposed full system to be investigated further is shown in Figure 23.

D. Online Velocity Estimation Learning

Leveraging a DCNN for velocity estimation opens up the possibility of performing online learning where the model is trained to correct for the drift over time. This could occur by feeding the model the error between the estimated velocity from the DCNN and the ground truth velocity obtained from the external positioning system measurements. This proposal is shown in the full system diagram in Figure 23. We credit Professor Robert Dick for this idea.

VII. Conclusion

Overall, we were able to investigate and obtain results for several modifications to indoor positioning. We found an average accuracy of 1.5 m for RSSI-based external position estimation, proving WLAN signals to be a lower cost alternative to other position systems with acceptable accuracy

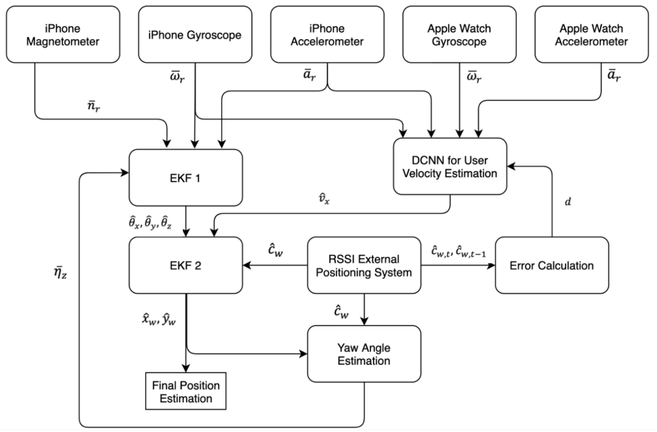

Fig. 23: Indoor position estimation block diagram for future work.

for the warehouse use case. We also explored the possibility of accurate velocity estimation using a DCNN model. While our model accuracy was low, further refining of the model may prove this to be a highly effective approach, and more importantly enable novel online learning of drift. The accuracy of the path localization was not increased when the Apple Watch position estimation from IMU data was averaged with that of the iPhone, however, we hypothesise that incorporating RSSI measurements for the watch may reveal an advantage with performing this multi-device sensor fusion.

ACKNOWLEDGMENT

We would like to thank Professor Robert Dick for his guidance and encouragement throughout this semester-long project. A fully remote semester to do the COVID-19 pandemic has challenged all of us in different ways, and we appreciate Professor Dick for his flexibility and generosity of his time and energy in spite of those challenges.

APPENDIX

Our code can be found here.

REFERENCES

[1] "About the Warehousing and Storage Subsector." U.S. Bureau of Labor Statistics, U.S. Bureau of Labor Statistics, www.bls.gov/iag/tgs/iag493.htm.

[2] A. Colombo, D. Fontanelli, D. Macii and L. Palopoli, "Flexible Indoor Localization and Tracking Based on a Wearable Platform and Sensor Data Fusion," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 4, pp. 864-876, April 2014, doi: 10.1109/TIM.2013.2283546.

[3] Y. Yuan, C. Melching, Y. Yuan and D. Hogrefe, "Multi-Device Fusion for Enhanced Contextual Awareness of Localization in Indoor Environments," in IEEE Access, vol. 6, pp. 7422-7431, 2018, doi: 10.1109/ACCESS.2018.2795738.

[4] X. Luo, W. J. O'Brien, C. Julien, "Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites," Advanced Engineering Informatics, vol. 25, pp. 355-363, 04 2011

[5] A. Shrestha and M. Won, "DeepWalking: Enabling Smartphone-Based Walking Speed Estimation Using Deep Learning," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647857.

[6] A. Golestani, N. Petreska, D. Wilfert and C. Zimmer, "Improving the precision of RSSI-based low-energy localization using path loss exponent estimation," 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), Dresden, Germany, 2014, pp. 1-6, doi: 10.1109/WPNC.2014.6843302.

[7] J. Niu, B. Wang, L. Shu, T. Q. Duong and Y. Chen, "ZIL: An Energy-Efficient Indoor Localization System Using ZigBee Radio to Detect WiFi Fingerprints," in IEEE Journal on Selected Areas in Communications, vol. 33, no. 7, pp. 1431-1442, July 2015, doi: 10.1109/JSAC.2015.2430171.

[8] H. Ding, Z. Zheng and Y. Zhang, "AP weighted multiple matching nearest neighbors approach for fingerprint-based indoor localization," 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS), Shanghai, China, 2016, pp. 218-222, doi: 10.1109/UPINLBS.2016.7809974.

[9] "RSSI: How to check RSSI value with NetSpot," NetSpot, 03-Oct-2019. [Online]. Available: https://www.netspotapp.com/what-is-rssi-level.html: :text=RSSI%20indicates%20the%20power%20level,zero%20usually%20means%20better%20signal. [Accessed: 21-Apr-2021].

[10] B. Horn, K.P. Indoor Positioning Using Time of Flight with Respect to WiFi Access Points, people.csail.mit.edu/bkph/ftmrtt_intro.

[11] D. Campbell. "Indoor Positioning with WiFi RTT and Google WiFi." Medium, Medium, 4 Oct. 2019, medium.com/@darryncampbell_83863/indoor-positioning-with-wifi-rtt-and-google-wifi-a638f1147b84.

[12] S. Warwick. "Developers Can Now Take Advantage of Apple's U1 Chip." IMore, IMore, 24 June 2020, www.imore.com/ios-14-developers-can-now-take-advantage-u1-chip-thanks-new-nearby-interaction-framework.

[13] U. Ahmad, K. Poon, A. M. Altayyari and M. R. Almazrouei, "A Low-cost Localization System for Warehouse Inventory Management," 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2019, pp. 1-5, doi: 10.1109/ICECTA48151.2019.8959774.

[14] N. Kothari, B. Kannan, E. D. Glasgwow, M. B. Dias, Robust Indoor Localization on a Commercial Smart Phone, Procedia Computer Science, Volume 10, 2012, Pages 1114-1120, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2012.06.158.