Location-Aware Smart Campus Security Application

Kaikai Liu, Navjot Warade, Tejas Pai, Keertikeya Gupta
Computer Engineering Department
San Jose State University (SJSU)
San Jose, CA, USA Email: {kaikai.liu, navjotchandrakant.warade, tejas.pai, keertikeya.gupta}@sjsu.edu

***Abstract*—One of the biggest challenges that Universities face today is the safety of its people on campus from crimes like mugging, battery and even shooting in or around the campus area. Using SJSU campus as an example, over 50 alert cases of burglaries, thefts, batteries, sexual assaults and other incidents have been reported in and around the SJSU campus over the last year. We have Bluelight emergency telephones placed all over the campus, in all buildings, elevators and on the campus grounds. These phones can be used to report emergency situations, suspicious activities, request escorts etc. However, there is a huge delay between the occurrence of incidents and the arrival of the policeman at the site.**

There is a critical need for a system that would allow the authorities to locate victims and respond faster to these incidents. To reduce the delay in reporting incidents and their occurrence time, we have developed a mobile application that will let users send alerts along with their real-time location to the UPD directly from their mobile phones. However, finding the position of a victim in a building is the most important challenge we are facing. Many existing systems do not work in indoor environment, and the state-of-the-art localization systems are either inconvenience to use or inaccurate enough to pin-point user's locations inside the building. In this paper, we propose a fine-grained location-aware smart campus security systems that leverages hybrid localization approaches with minimum deployment cost. Specifically, we effectively combines the Wi-Fi fingerprinting localization approach with the Bluetooth beacon based trilateration approach, and improves the location accuracy to the meter-level with low cost.

***Index Terms*—Indoor localization, smart campus, location alert, fingerprinting, trilateration**

I. Introduction

Over the last year, more than 50 alert cases of burglaries, thefts, batteries, sexual assaults and other such incidents have been reported in and around the SJSU campus, with more than twenty of these being reported in the last two months. This has affected the reputation of SJSU adversely and questions have been raised about the current security structure of our university.

We have Bluelight emergency telephones placed all over the campus, in all buildings, elevators and on the campus grounds. You can find these telephones in all buildings below blue colored light, in all elevators and on the campus grounds as tall blue poles with blue lights on them. These phones can be used to report emergency situations, suspicious activities, request escorts etc. The University Police Department (UPD) also allows you to report non-emergency situations and anonymous tips at phone number or email address provided at their website. However, there is a huge delay between the occurrence of incidents, the reporting time and the arrival of the UPD at the site. By the time the officials reach the victims, the suspects have already fled the scene without facing any consequences.

To solve the problem of the delay in reporting the incidents and the actual occurrence of incidents, we have decided to develop a mobile application that will let users send alerts along with their real-time location to the UPD directly from their Android mobile phones. In the mean time, we can use the mobile application to cover areas that without Bluelight phones. When a user sends an alert message through our app, the UPD will get the current location of the victims phone along with previously stored information such as profile picture, name, age, and others. The UPD will also be able to monitor their real-time location on the web page and once the issue is resolved, the tracking will be turned off. With this app, we aim to significantly reduce delay in reporting incidents, thus allowing the victims to receive timely assistance and cut the crime rate in and around SJSU campus.

However, finding the position of a victim in a building is the most important challenge we are facing. In the last few years, Location Based Services (LBSs) [1] have seen a boom in their demand. LBSs can range from services provided within countries to services provided within a few meters of a beacon. In this era of mobile phones, LBS has seen increasing demands and has opened doors to new possibilities limited only by your imagination. One of the major component of LBS technology is the localization algorithm, which impacts the performance, reliability of LBS systems and the battery life of your mobile phone. Many existing systems do not work in indoor environment, and the state-of-the-art localization systems are either inconvenience to use or inaccurate enough to pin-point user's locations inside the building. Some of the challenges with IPS are effects of obstacles on the signal strength, movement of subject, interference [6] from other sources of signal, and line of sight nature of many positioning techniques.

One of the biggest motivation of our project was that existing Wireless network is excellent and spread throughout the campus. With the help of the wireless network, we can implement an indoor positioning algorithm (in this case, Fingerprinting) and find the real-time position of users through their phones. In this paper, we propose a fine-grained location-aware smart campus security systems that leverages hybrid localization approaches with minimum deployment cost. Specifically, we effectively combines the Wi-Fi fingerprinting localization approach with the Bluetooth beacon based

trilateration approach, and improves the location accuracy to the meter-level with low cost. We compared our system with three existing systems: Easy Floor Map, Redpin and FIND, the result shows that our system achieves best accuracy with no performance degradation over time.

II. System Overview

A. System Design

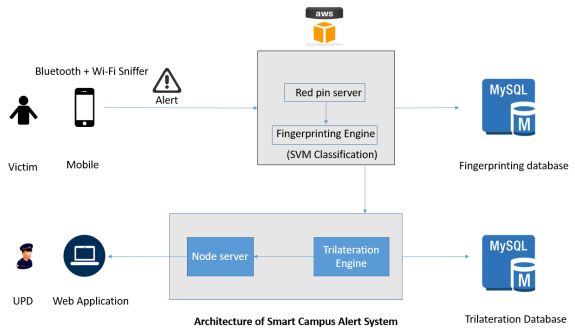


Fig. 1. The system architecture.

Fig. 1 shows the architecture of our proposed Smart Campus Security Alert System, which mainly consists of Mobile Application, cloud server, fingerprinting engine database and trilateration engine, and the Web Application.

We have implemented several modules which perform different functions needed in an alert system. The servers are deployed on the AWS platform to have independent access from any location. Our system also contains a mobile application and the web application which are deployed on mobile and web platform.

The main parts of our architecture are fingerprint recording: the fingerprints for all the location inside the campus buildings are recording in the first phase. These fingerprints are stored in the cloud database. These fingerprints are sent to the database at constant intervals to create large set of values which will be helpful to calculate the location.

Locating the user: As soon as the mobile user presses the alert button. The mobile application will activate the Bluetooth and the Wi-Fi scanner. These scanners will analyze and record the all the signal strengths of the nearby access points and will send these measurements to server. The server will calculate the position of the user using these measurements and will send this location to Mobile and the Web application respectively

Web application: The web application serves the purpose of receiving the notifications and alert from the mobile application which is used by the victim to send the alert. It also provides a clean user interface which shows the location of the user along with the alert details and the user details. This web application will inform the security officials about the location of the victim and will constantly change as the user moves through the area. This keeps the security officials aware of the current position of the user.

Server and database: There are two cloud servers- Node and Java. These are deployed on AWS cloud which makes the servers accessible via the internet. With the help of this server the fingerprints from any building can be recorded. Hence, each individual inside the campus can send the fingerprints as per their convenience. Therefore, the training model will be constantly trained and this trained model will be used for predicting the location of the user when the user sends the alert.

B. Indoor Localization via Wi-Fi systems

Using Wi-Fi systems for indoor positioning has a number of advantages. Some of these advantages are listed: Wi-Fi is readily accessible in almost all buildings nowadays. However, the advantages of Wi-Fi networks provide a cost-effective solution for implementing LBS indoor environments [1]. No additional hardware is required. No need to install further software, Wi-Fi networks offer high scalability. Unlike GPS, Wi-Fi signals can penetrate through obstacles, fingerprints for target locations are available for most indoor positions. Despite these advantages, there are a few drawbacks of using Wi-Fi systems. For example, a lot of manual work is required to acquire fingerprints of the entire area of coverage. The presence of obstacles causes multi-path and the environmental conditions (climate, etc.) all affects the signal strength. Other Wi-Fi devices might lead to interference which further reduces accuracy.

In Wi-Fi fingerprinting, we create a map of a selected area based on the Received Signal Strength Index (RSSI) values of available Wi-Fi access points. Depending on the requirement of the fingerprinting algorithm, some filtering may need to be applied to collect the fingerprint values of particular access points only. These values are then stored in a special database called Fingerprint Database. The mobile device is located by taking measurements at a location and matching the fingerprint with the values in the database [15]. There are two phases in fingerprinting. In the first phase is an offline phase, i.e. there are no database values yet and the system needs to be trained. This phase is known as the training phase or calibration phase. Radio maps are recorded in this phase and fingerprint values of all access points are entered into the database. In the second phase, the fingerprinting algorithm takes the user input and produces an output after matching the input with the database entries we obtained in the first phase. Fingerprinting algorithms use usually use some pattern recognition techniques such as K-nearest Neighbors (KNN), artificial neural networks, Bayesian interference or support vector machine (SVM) [2]. The system may also use a combination of these algorithms to improve accuracy. Fingerprinting may also be combined with other techniques such as trilateration as shown in [3].

III. System Implementation

We have added enhancements to the original Redpin server [4]. This server is used to communicate with the database engine and the SVM engine. We have kept the functionality to communicate with the database for Wi-Fi readings. We

have added the functionality to add Bluetooth readings. The readings sent by the scanner are to be added to the database. The readings dataset is divided into individual readings and then added to the Bluetooth readings table created in the database.

Fingerprinting Engine. Whenever the fingerprinting engine receives the values of the alert message it uses the SVM model to predict the location of the user and the map. This data is then forwarded to the trilateration engine for further calculation. Fingerprinting engine also works to create a SVM model in the training phase.

Node Server. Once the fingerprinting engine calculates the room number it is sent back to this server. This server then sends the data to the node server for further positioning and display on UI.

Fingerprint Database. The fingerprint database stores the fingerprint measurements of the different locations that we have mapped in the system. It will be used by the fingerprinting engine to match and retrieve the user location.

Trilateration Engine. We are achieving trilateration of a user using Bluetooth devices fixed by us at various position. With an increase in distance between the Bluetooth devices the signal strength decreases. But this signal strength also depends upon other environmental factors like solid obstacles, walls, people etc. Each environment and device differs. We have used Bluetooth devices as our nodes.

Trilateration Database. Trilateration database is used to map the locations to the Bluetooth devices corresponding to it. This table is used by the trilateration engine for localization. When the trilateration engine receives the name of the Bluetooth from this database it will filter out the rest of the Bluetooth nodes found by the Bluetooth sniffer. This will help the trilateration engine calculate the position better by removing the unwanted Bluetooth nodes

Web UI. The web UI is used only by the authorities to see the location of the user sending the alert. It displays the map with the user location on it. The user position is also displayed in the text format beside the map.

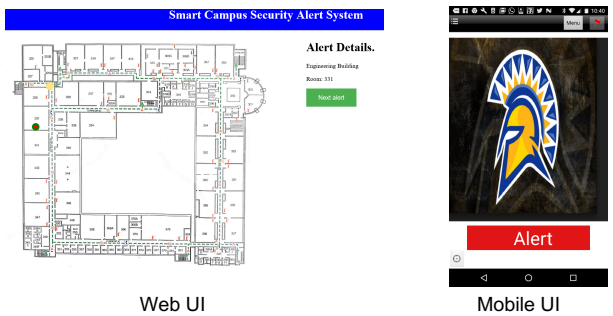


Fig. 2. The developed system consists of the web part and mobile app part.

Fig. 2 shows the developed system consists of the web part and mobile app part. Fig. 3 shows the architecture of the classes of our implemented system.

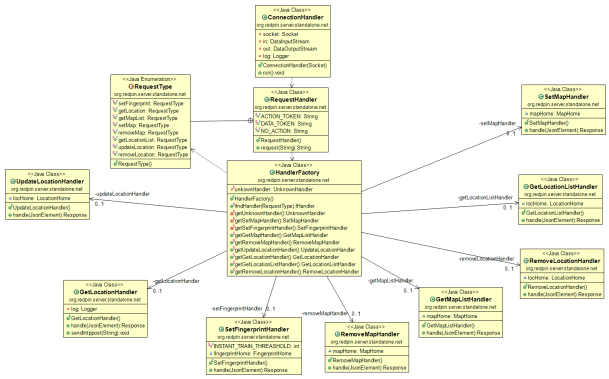


Fig. 3. The architecture of the classes of our implemented system.

IV. Indoor Localization

The implementation for the Smart Campus Indoor positioning system covers the process setting the fingerprints to the database which will help build the training model for predicting the location of the mobile user. It also covers the positioning using Bluetooth trilateration mechanism which give more accurate position of the mobile user. It consists of three major components Android application to record and send the fingerprints to databases, Java Server to send the data to database and calculate the location of user using SVM classifier. Node server to perform trilateration to get the approximate location of user. Web Application to show the results of SVM classifier and Bluetooth trilateration on the map. Fig. 4 shows the sequence diagram of our system.

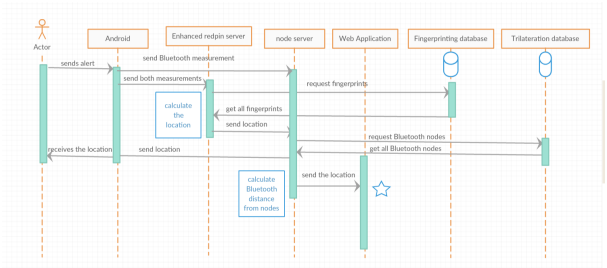


Fig. 4. The sequence diagram.

First, the Sniffer (This is the scanner which is also referred to as sniffer) will be initiated once the user wants to take a reading. It will create a sniffer object and initialize the variables. The sniffer will then start to scan the Bluetooth and Wi-Fi access points. The main attributes it will scan are their IDs, Names, and RSSI values. These values are then sent through filters to clean out unwanted data. For Wi-Fi only SJSU Premier Access points will be taken into consideration. Other Wi-Fi will be removed from the sniffer object. Similarly, for Bluetooth only the Bluetooth devices that we have positioned will be kept in the sniffer object. Rest of the Bluetooth device like phone and speakers nearby will be removed from the sniffer object. Once the sniffer object is cleaned the sniffer object will be sent to the Backend server.

Then either it will be added to the database for storage or to the Fingerprinting engine to calculate the position.

1) Fingerprinting Implementation: The setting of fingerprint consists of sending the signal strengths of Bluetooth and the Wi-Fi access points to the server. Both Wi-Fi and Bluetooth service are registered and bind to the background service of the mobile application. This keeps the Wi-Fi and Bluetooth service running in the background such that even if the application is minimized the data will be sent to the server at fixed interval. The two sniffers will capture the signal strengths of all the nearest Access Points and will send it to the server.

The two sniffers will capture the signal strengths of all the nearest Access Points and will send it to the server. A handler is set to handle the response and failure of the action of adding the fingerprint to the data base. It will return the success message if the fingerprint tis set successfully and will execute On Failure method if the request does not succeed in adding the fingerprints to the database. We have set several finger prints across the floors of the SJSU engineering building to setup our fingerprint database. This make us to physically go to the location which we want to set and take measurements.

While setting up the fingerprints we had to take care that the fingerprints are not too near or far away from each other. Therefore, most of the fingerprints we have taken are either 20 to 40 feet away or in a line of sight with each other. This implementation gave us more correct readings than setting the fingerprints too near to each other yet useful when locating the user's position. After sending all the location and the fingerprints to the database, the fingerprinting engine will be able to retrieve these values and then the SVM algorithm would be able to create the train model. This model can be used for predicting the location of the users for sending the alert.

2) Support Vector Machine Implementation: The location of the user is determined with the help of Support Vector Machine algorithm [5] which predicts the location of the user based on the data which is provided by the train model which was generated during the fingerprinting. Libsvm 2.9 [5] is used as a java library to classify the locations in the training model. Libsvm has various tools for C-SVC classification, epsilon-SVR regression and one-class SVM. SVM implementation can be divided into two parts:

Training. All the fingerprints are pulled from the database and are categorized into the numeric form such that each location and their corresponding Access points will represent distinct numeric value. Vertical blue line represents the id of the location starting from vertically. Horizontal red line represents the id of the Access points (Wi-Fi + Bluetooth) starting horizontally. The fingerprint data is transformed to SVM format before it is used for training the model.

Fig. 5 shows the fingerprinting training flow chart.

Prediction. With the help of the training model, it is possible to predict the location of the user using the predict function provided by the libsvm library. Test fingerprints are recorded using the android Wi-Fi and Bluetooth scanner and are passed to predict function as an input. The libsvm provides

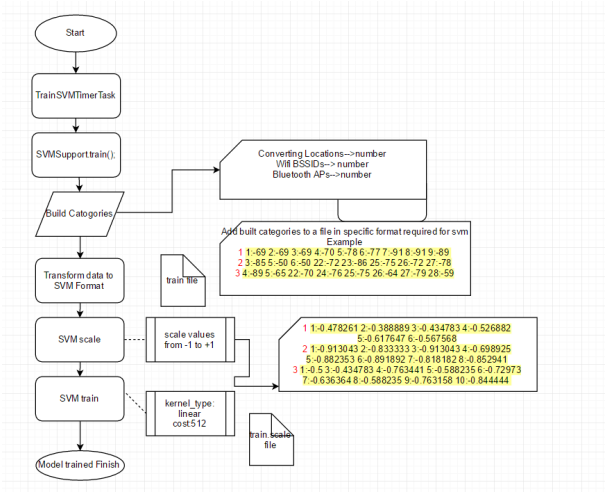


Fig. 5. The SVM training flowchart.

the functionality to predict the values of attributes based on the previous set of attribute values. More the values are recorded for the training model, more the accuracy given by the prediction. The test data would be sent to SVM which will contain the signal strengths of the various access point. Based on this data, the prediction model will find out the name of the given location. For best results of the prediction, the SVM algorithm is tweaked by making few changes. Fig. 6 shows the SVM prediction flow chart.

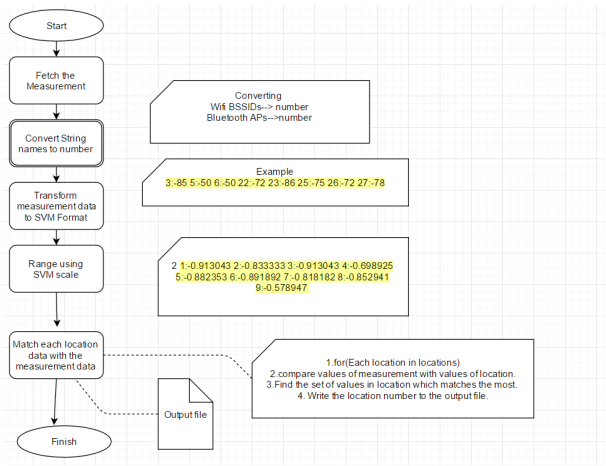


Fig. 6. The SVM prediction flowchart.

A. Bluetooth Trilateration

Average signal strength attenuation. Fig. 7shows the average signal strength column is plotted against the distance column. It is seen that the averaged signal strength decreases as the distance increases. These distances along with their RSSI values were recorded in the line of sight. The averaged values were more accurate as compared to normal RSSI values (without averaging). The graph is plotted with 3 measurements at each 10 feet distance. The distance of 60 feet was covered

which is more than the distance of the average size hall. The average RSSI values recorded were more accurate the next time the readings were taken.

To obtain the distances based on measured RSSI, distance table was made based on the Bluetooth signal attenuation. The average signal strength column shows the values of the RSSI averaged at a fixed position. We have measured the RSSI values at intervals of 3 feet. From the average signal strength graph, it can be concluded that as the distance increase by 10 feet each time the signal strength drops by -5 dm. For example, if the frequency received from a node is between -45 dBm to -52 dBm, the distance of that node from the mobile device is approximately 3 feet. Similarly, if the averaged value of the RSSI falls between -59 dBm to -68 dBm, the distance will be 15 feet approximately.

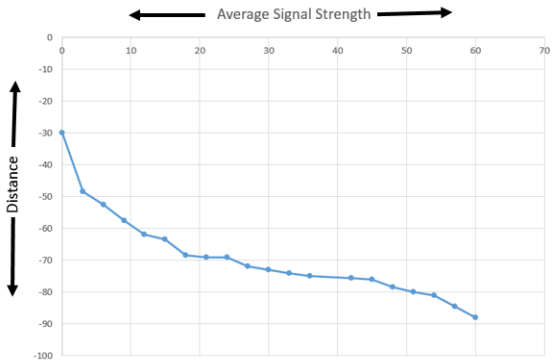


Fig. 7. The average signal strength attenuation.

We are achieving trilateration of a user using Bluetooth devices fixed by us at various positions. With an increase in distance between the Bluetooth devices the signal strength decreases. But this signal strength also depends upon other environmental factors like solid obstacles, walls, people etc. Each environment and device differs. In this project, we used three Bluetooth nodes for trilateration. We placed these three nodes in a room large enough for calculating the positions of the mobile device in multiple positions. The Bluetooth nodes transmit signal the mobile phone receives from each of them and shows their respective RSSI value.

The trilateration engine calculates the distance of the user from each of the Bluetooth node. show its position by plotting circles of the calculated radius on the floor map of the building, with the respective Bluetooth nodes being the center of the circles. The intersection of all the circles gives the approximate position of the mobile phone. Note that the calculated position is an approximate value because the intersection of the circles is not a pinpoint and larger than the size of the mobile phone. Fig. 8 shows the localization database including fingerprinting and trilateration.

V. Performance Evaluation

To evaluate the performance of our proposed system, we compared the results of the four types of solutions. Each

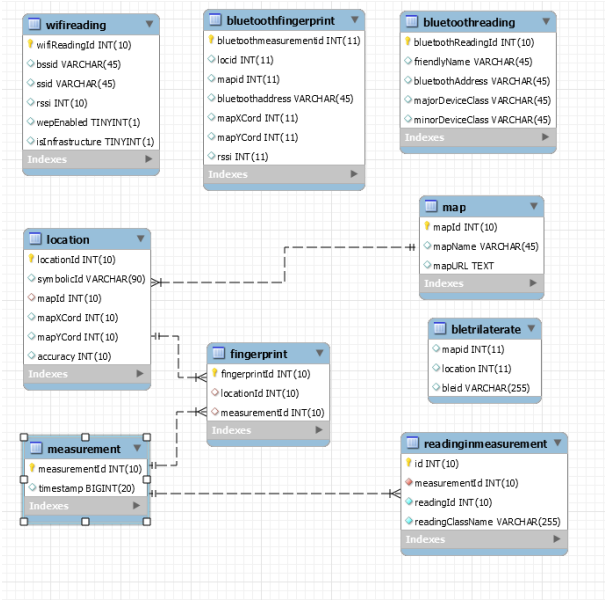


Fig. 8. The localization database including fingerprinting and trilateration.

solution takes input of a particular type of data set. Training data was collected by each solution. The same location points were selected as training data points for each solution in order to get the fair comparison. Currently as we are testing on one floor we have taken around 30 readings as training data on the floor. The actual database covers two floors. The detailed descriptions of different solutions are:

1) **Solution 1:** All Wi-Fi access points were taken into consideration. SVM model was created on basis on them. Examples of these systems are Redpin, Find, easy floor map as open source projects.
2) **Solution 2:** Some low-fidelity access point has been removed with only premier access points taken into consideration. SVM model was created on basis of them.
3) **Solution 3:** Bluetooth RSSI readings were taken into consideration. SVM model was built on basis of both readings (Wi-Fi and Bluetooth).
4) **Solution 4:** Our proposed solution. We utilize model-based trilateration to further improve the fingerprinting performance.

There are three types of tests for IPS. One is for floor level testing using fingerprinting. In this test, we will compare solution 1 and solution 2. We have not considered solution 3 and solution 4 as we did not add Bluetooth nodes in this test. The performance of solution 2, 3 and 4 should be the same. The second test is room level testing in which we will compare all four solutions. In this case we are testing all four solutions to compare their accuracy at room level, i.e. to see which one works best within a room. We have conducted the room test in Engineering Building Room 331. The third test compares solution 1, 2 and 3 with respect to their aging, i.e. how much the accuracy deteriorates over time. For this test we took measurements of solution 1, 2 and 3 at the same

test locations after 24 hours and then again after 7 days. We calculated the average distance error of all three sniffers for each day's readings and compared them with respect to the change in accuracy of each. We did not consider solution 4 in this test because the solution 4 and solution 3 should have the same deterioration performance.

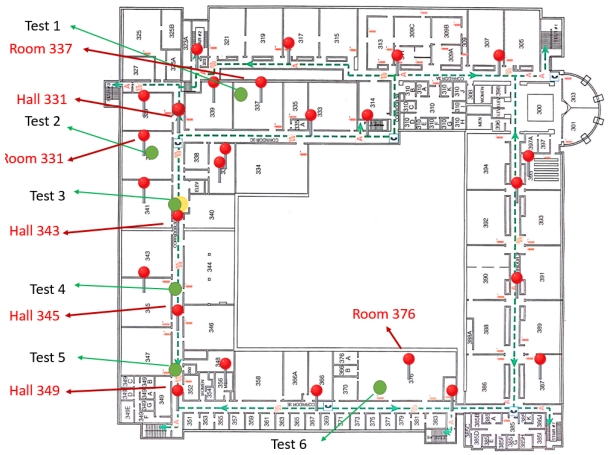


Fig. 9. The test locations in the third floor of our engineering building.

Fig. 9 shows the test locations in the third floor of our engineering building. The red markers in the figure shows the fingerprint locations, i.e., the training points. The Green markers are the test points used to evaluate the accuracy of different solutions.

When we need to test the accuracy within one room, we deployed the Bluetooth node in Room 331. As shown in Fig. 10, we selected multiple points at various locations in the room as fingerprinting point and test point. This is the same room where we are going to do the trilateration. The green circles in Fig. 10 show the actual location of the device of each test. For solution 4 we have taken multiple tests at each location. The red dots show the location of the Wi-Fi fingerprint. The black dots show the location of the Bluetooth nodes in the room.

A. Floor Level Testing

Fig. 11 shows the location error (in feet distance) of the multiple testing points for solution 1; Fig. 12 shows the location error (in feet distance) of the multiple testing points for solution 2. Fig. 13 shows the performance difference between the solution 1 and solution 2 at the floor level. The average distance error is calculated from the readings taken for solution 1 and solution 2 as shown in Fig. 9. From the results, we see that solution 2 gives less average error in distance when compared with solution 1.

B. Room Level Comparison for all Solutions

We have tested all four solutions at room level with multiple test locations as shown in Fig. 10. We are comparing two aspects of these systems: the average distance error which is the distance between the calculated distance and the actual

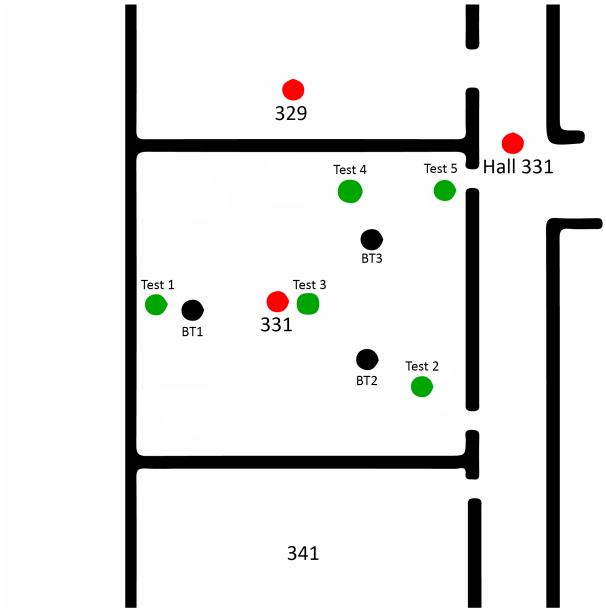


Fig. 10. The test locations in Engineering Building Room 331.

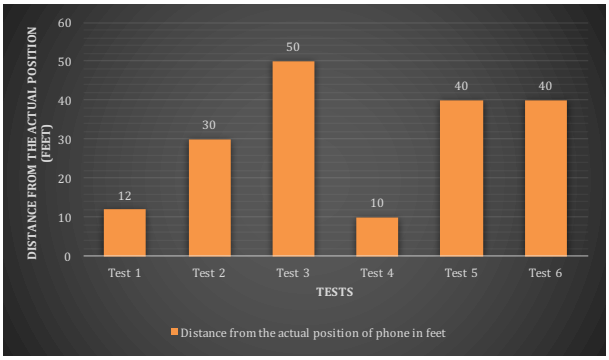


Fig. 11. The location error in distance (feet) for solution 1.

distance and the accuracy in terms of the displayed room of the user. For example, if the user is in Room 331, the displayed room should also be Room 331 for the result to be correct.

From Fig. 14, we can see that solution 4 (our proposed solution) has a significantly low average distance error. It provides more accurate position of the user on the map.

Fig. 15 shows solution 3 and solution 4 have 100 percent accuracy when it comes to the room level accuracy. Solution 1 being poorest, and solution 2 is better but the accuracy is still not acceptable. We can conclude that our proposed solution (solution 4) has the best localization implementation as it has a 100 percent accuracy and least average distance error.

C. Accuracy over long time period

The key problem for fingerprinting-based solutions is the performance degradation over time as the radio signal fingerprint changes over time. As shown in Fig. 16a, we are comparing sniffers solution 2 and solution 3 for accuracy over a period of time. We tested solutions at day 0, after 7 days and again after 11 days. Day 0 is the day on which we took

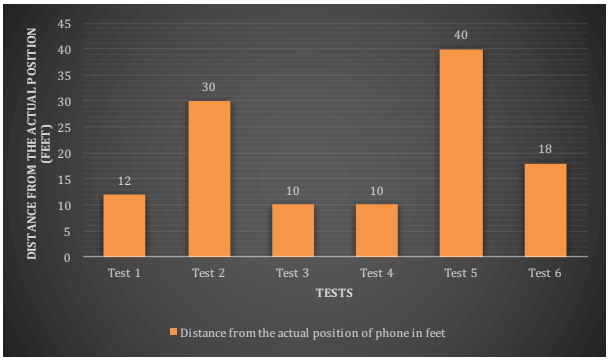


Fig. 12. The location error in distance (feet) for solution 2.

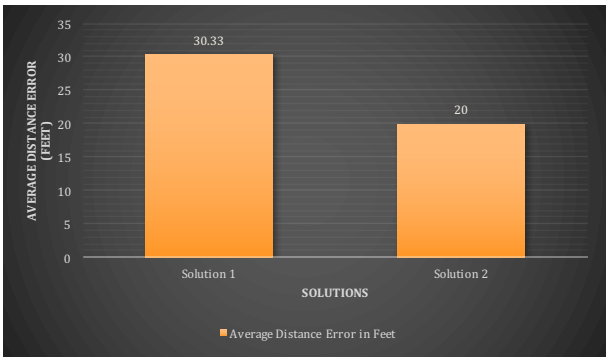


Fig. 13. The floor-level location error comparison in distance (feet) for solution 1 and 2.

the readings of the fingerprint. We can see that both solution 2 and 3 had the same results on Day 0, which were all correct. However, after a week, solution 2 starts to show incorrect results. It's average distance error increases and accuracy decreases. On the other hand, solution 3's average distance error and accuracy remain constant over a period of time. This means that solution 3 produces reliable results, which is a highly desirable property in any localization system.

We see that solution 3 has a constant distance error over a period of time. On the other hand, solution 3's average distance error and accuracy remain constant over a period of time.

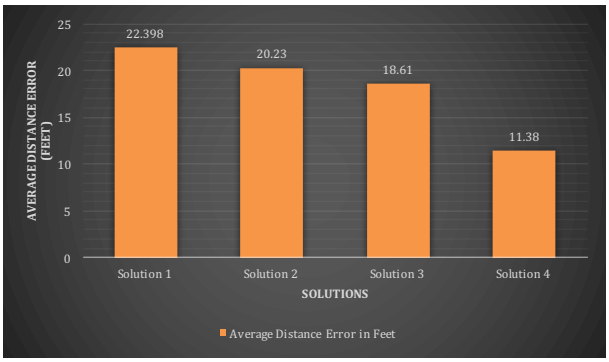


Fig. 14. The room-level comparison of average location error comparison in distance (feet) for all solutions.

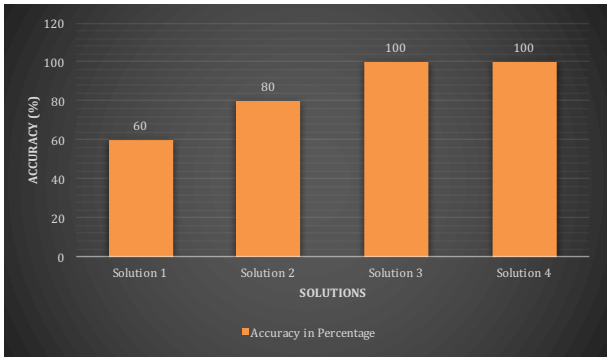
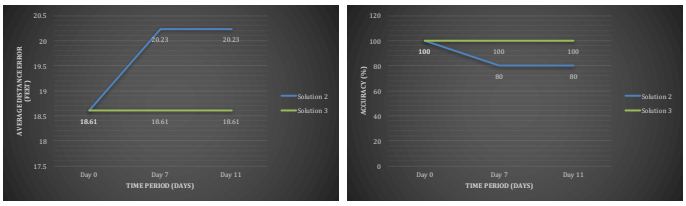


Fig. 15. The room-level comparison of localization percentages for all solutions.



(a) Figure 1 (b) Figure 2

Fig. 16. The performance degradation overtime: (a) accuracy in terms feet; (b) accurate localization percentage.

This means that Sniffer 3 produces reliable results, which is a highly desirable property in any localization system. This is contributed by the added Bluetooth node as the reference node, which can mitigate the performance degradation problem. However, solution 2 has degraded over time and its average distance error has increased.

From the above results, we can conclude that using a hybrid system (in our case Fingerprinting plus Trilateration) results in a more stable and reliable localization system.

VI. Related Work

Redpin is an open-source, fingerprinting based indoor positioning system built specifically for mobile devices, and relies mostly on the user community to enter the location values rather than the conventional training and setup phases that most of the other systems use. This is in order to reduce setup time and allow users to use the application right away. Through the fingerprinting algorithm, Redpin guarantees room-level accuracy [6].

Redpin provides symbolic identifiers to determine the location of a device. For example, Redpin will save the building name and room number to identify a particular room (Engineering Building, Room 189) instead of using geological co-ordinates. This allows Redpin to eliminate the training and usage phases and adjust to changes - like downtime of an access point or replacement of a router - with ease. Disadvantages of Redpin- No support for positioning with Bluetooth, Does not use trilateration. Redpin gives only room-level accuracy. To track the real time position of the users mobile device, trilateration can be used.

FIND is similar to Redpin and it also depends upon its user community to insert data into the database. It also uses the same algorithm of SVM as Redpin [6], [7]. Find uses posteriors Nave-Bayes to calculate the location of the user. It is a classification technique based on Bayes theorem with an assumption of independence among predictors. It takes the fingerprint and the parameter set. Using this fingerprint, it also passes a parameter set which is the schema of the fingerprint. This schema includes the data of the access point in the form of key, value of pair.

Easy floor map is an indoor grid based localization model, which locates the user location on the principle of cell grading [8], [9], [10]. Similar to other indoor localization techniques the Easy Floor Map contains two phases training and locating. It records the Wi-Fi access points in the training phase and grades the cell based on the similarity of the test data. In learning mode, the map is loaded and the map is divided into a grid of size 10 X 16. Each cell in the grid represents a distinct specific location and is represented by a square. The scan rate is set to scan the for the Wi-Fi access points at the fixed interval. This scan rate could be set to maximum to get more accurate results.

VII. Conclusion

With the help of this project, victims of incidents such as theft, battery, etc. will be able to get timely help from the UPD. This will help reduce crime in and around the SJSU campus and also receive timely response for the victims. This application can also be further used to as a positioning system in other applications. From our observations of the positioning system, we found out that using a hybrid localization system such as ours (where we used Wi-Fi + Bluetooth for Fingerprinting and Bluetooth for Trilateration) gives better localization results than using just a single technique such as Fingerprinting or Trilateration alone.

Acknowledgment

The work presented in this paper is supported in part by National Science Foundation under Grant No. CNS 1637371.

References

[1] A. Taheri, A. Singh, and A. Emmanuel, "Location fingerprinting on infrastructure 802.11 wireless local area networks (wlans) using locus," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 676–683.

[2] Y. Liu and Z. Yang, *Location, localization, and localizability: location-awareness technology for wireless networks*. Springer Science & Business Media, 2010.

[3] S. Chan and G. Sohn, "Indoor localization using wi-fi based fingerprinting and trilateration techiques for lbs applications," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, p. 4, 2012.

[4] P. Bolliger, "Redpin-adaptive, zero-configuration indoor localization through user collaboration," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM, 2008, pp. 55–60.

[5] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[6] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[7] C. Koweerawong, K. Wipusitwarakun, and K. Kaemarungsi, "Indoor localization improvement via adaptive rss fingerprinting database," in *Information Networking (ICOIN), 2013 International Conference on*. IEEE, 2013, pp. 412–416.

[8] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: indoor pathway mapping made easy," in *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2013, pp. 85–98.

[9] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[10] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1031–1043, 2015.