Indoor Positioning Framework for Most WiFi-Enabled Devices

Yucheng Huang
University of California, San Diego
yuh063@eng.ucsd.edu

Yuankai Le
University of California, San Diego
y2le@ucsd.edu

Wenhao Sheng
University of California, San Diego
wesheng@eng.ucsd.edu

Abstract

Nowadays, we have seen that cellphones and other mobile devices, such as laptops and tablets, are playing an important role in assisting and entertaining people. A huge part of people's social life involves cellphones. Not knowing where one's cellphone is could be a stressful thing. One might accidentally leave it in the kitchen, and it would cost her a few minutes to walk across the entire house to find it. Hence, it would be helpful to have an object-finding system that is able to localize an electronic devices.

1. Introduction

Indoor positioning systems (IPS) use sensors and communication technologies to locate objects in indoor environment. Position location of a user or device in a given space has drawn great attention because of the potential to leverage commercial applications such as advertisement and social networks (*A. K. Dey, 2001.*). Location detection in outdoor environment has been successfully implemented using GPS technology (*G. Xu, GPS, Springer, 2003.*). However, in indoor environments, the usability of the GPS is limited, due to the lack of line of sight and attenuation of GPS signals as they cross through walls. Hence, specialized methods and technologies for indoor positioning systems (IPS) are needed. An IPS estimates the target object location from the observation data collected by a set of sensing devices or sensors (*H. Koyuncu and S. H. Yang, 2010.*). Previous works in this field made use of optical technologies (infrared technology, visible light communication, etc), sound-based technologies (ultrasound, audible sound, etc), and radio frequency technologies (Wi-Fi, bluetooth, ZigBee, radio frequency identification, ultrawideband, etc). Our team decided to use figerprinting method to calculate approximate locations. The term has been used especially as a way to obtain locations from detection of Wi-Fi signals and the

like. Since these are registered at mobile devices. It consists of two phases: training and position determination. In the training phase, a radio map of observed signal strength values from different locations is recorded. In the position determination phase, the signal strength values observed at a user device are compared to the radio map values using proximity matching algorithms, such as the k-nearest neighbor algorithm, in order to inder current user location (*P. Jiang, Y. Zhang, W. Fu, H. Liu, and X. Su, 2015.*).

2. Overview of Method

This section gives an overview of the methods applied.

2.1. Floorplan Data

We struct our floorplan data in an flp file. Fig 2 illustrates the format that we write in the file. Each row of the data represents a single wall. We create 74 rows of data in the file to form the floor plan, which is shown in Fig 1. We do not use the first 4 columns of data, so we set them to zeros. The fifth column is the mean of the error, and the sixth column is the standard deviation of the error. Since the wall should has height, which is not shown in Fig 1, we need to record four cornors to draw a rectangular plane rather than record two points to draw a single line. From seventh to nineth column, they represent the value for the position on x-axis, y-axis, and z-axis for the first point respectively. From tenth to twelfth column, they represent the value for the position on x-axis, y-axis, and z-axis for the second point respectively. From thirteenth to fifteenth column, they represent the value for the position on x-axis, y-axis, and z-axis for the third point respectively. From sixteenth to eighteenth column, they represent the value for the position on x-axis, y-axis, and z-axis for the forth point respectively. Unfuortunately, we do not know how to construct a 3D-model in MATLAB. Thus, we only use the value of x-axis and y-axis to draw a line to represent a wall, and we will set the value of z-axis to zero, when we create the test point on the map. The unit

Figure 1. Our Floorplan Layout
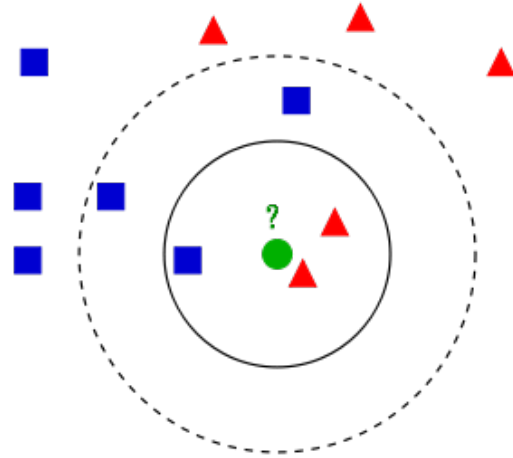

Figure 2. Our Floorplan Data

Figure 3. Example of classification rule with k-NN model.

3. Results

Figure 4 is the floor plan with labeled Room Number, and we are going to use it as reference to check if the result we obtained is correct or not.
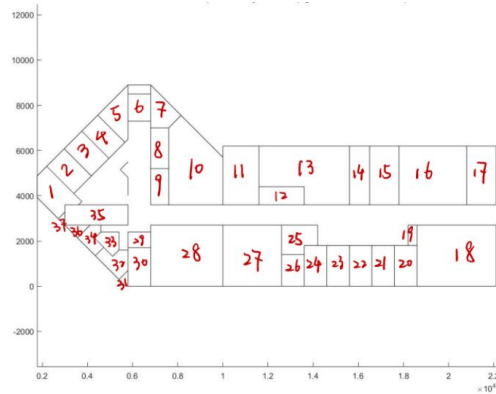

Figure 4. Floor plan with Labeled Room Number

for the value of the coordinate will be pixels, and we will use 100 pixels to represent one meter in the real world.

2.2. Location to Signal Strength Transformation

Knowing the relative distance from the object to the receivers in each room, we would like to convert this distance to Wi-Fi signal strength. To measure the signal strength, we use the free space path loss (dBm), which depends on two parameters: first is the frequency of signal; second is the wireless transmission distance. The relationship goes like: FSPL (dBm) = $20\log10(d) + 20\log10(f) + K$, where d is the transmission distance in kilometers, f is the frequency in MHz (we use 2.4MHz), and K is a constant that depends on the units used for d and f. K is equal to 32.44.

2.3. K-Nearest Neighbor

K-NN algorithm is a non-parametric method used for classification and regression. In our project, we will focus on the k-NN classification. In k-NN classification, an object is classified by a majority vote of its neighbors. For each HOG feature data points. The k-NN algorithm is going to obtain the classes of its k closest neighbors. (Using the Euclidean distance as sole criteria), and assign the class of query image to the majority of classes. For the purpose of this project, we set k=37. Fig 3 illustrates the process.

By running the code in Matlab, it will print out a floor-plan map as shown in figure 1 to us. Then we can choose a point by right-clicking somewhere on the map which will be shown as a star symbol in the floor-plan. The program will return the room number of the room where the point we choose locates in. The result will be stored in the variable called class-obj.

Firstly, we want to check whether the program will give us the correct room number if we want to find an object in somewhere near the center of the room.

Trial 1: We choose a point on the floor plan as shown in figure 5, and the result gives that the point locates in Room 10. Compared with the actual room number shown in figure 4, the result is correct.
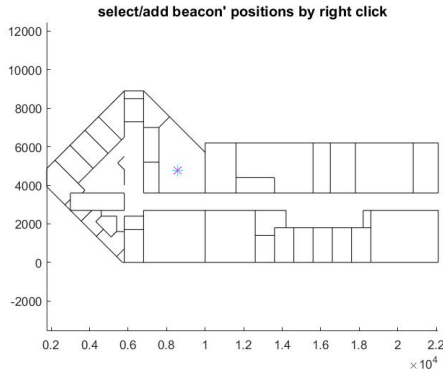
Figure 5. First Trial

Trial 2: Same as before, we choose a random point on the floor plan. In this case, the point chose is shown in figure 6, and the result gives that the point locates in Room 3. Compared with the actual room number shown in figure 4, the result is correct.
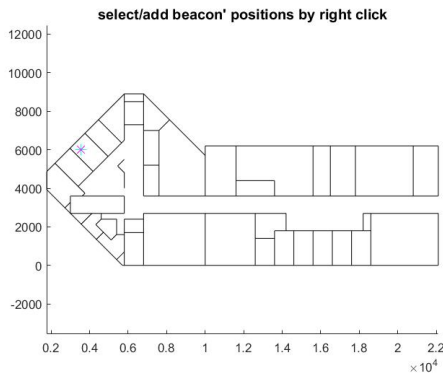

Figure 6. Second Trial

Trial 3: Point chosen is shown in figure 7. Return result is room 33 which is correct by comparing with the actual data in figure 4.

Therefore, the program works fine when the check point is near the center of some rooms.

Then, let's check if the program can work properly if we want to find a point on the corner room.

Trial 4: Point chosen is shown in figure 8. The program shows that the is in room 37 which is not the correct since the reference room number for this point is 1.

Trial 5: Point chosen is shown in figure 9. The program shows that the point is in room 33 which is not the correct since the reference room number for this point is 32.

Trial 6: Point chosen is shown in figure 10. The program shows that the point is in room 4 which is also different from the one in figure 4. The reference room for this point should be 5.
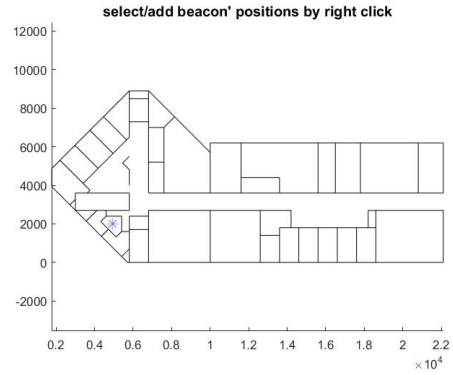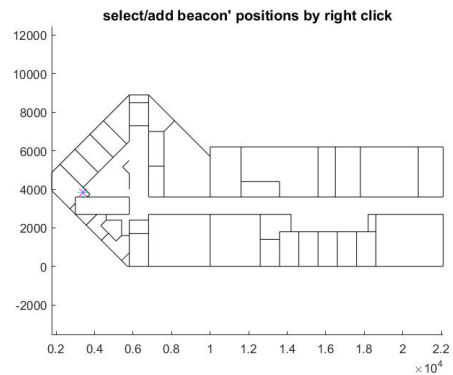

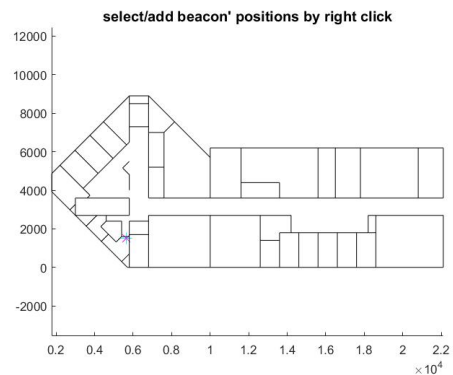Figure 7. Third Trial

Figure 8. Fourth Trial

Figure 9. Fifth Trial

Trial 7: Point chose is shown in figure 11. The result is 12 which is the correct result since it is same as the reference true result shown in figure 4.

Trial 8: Shown in figure 12. The result is 35. Wrong! It should be 36.

Therefore, the program sometimes can not give us the correct corresponding room number if we want to find a point in the corner of the room.
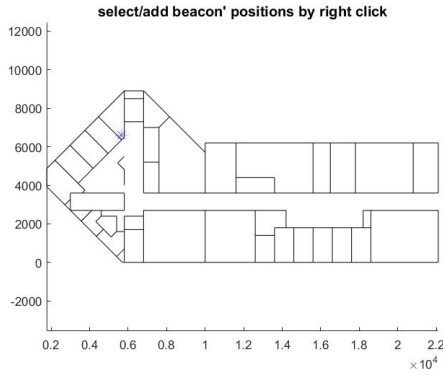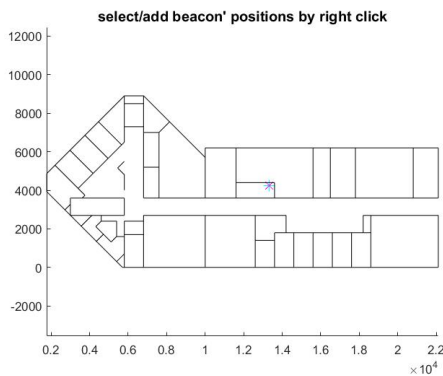
Figure 10. Sixth Trial
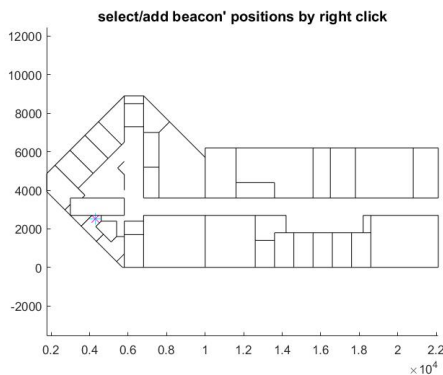

Figure 11. Seventh Trial

Figure 12. Eighth Trial

The reason for that is because the k-NN algorithm used Euclidean distance as sole criteria, which may assign the room number of neighbor room to the point if the point is very close to that neighbor room, especially for some special case where the shape of room is very complex and the size of it is very small.

4. Conclusion

In this project, we successfully implemented the Indoor positioning systems on finding an object. In the sense of physical application, the way we obtain the objects' location is detecting the WI-Fi signals. In the sense of machine learning, we used K-Nearest Neighbor method to find the best result. However, our project is still able to be improved a lot. As is shown in result section, when the point we chose is at the corner of some room. The program will give us the wrong answer. Therefore, to improve this project, work further and try/apply other algorithm on the project is very important. With this system, we are able to localize the electronic devices within a small area. It can also be used in the factory or storage where the GPS technology can't find the location of the object very accurate. When GPS technology works with a very large range, IPS technology will solve the small range localization problem.

References

[1] Salamah, Ahmed H., Mohamed Tamazin, Maha A. Sharkas, and Mohamed Khedr *An Enhanced WiFi Indoor Localization System Based on Machine Learning, 2016.*

[2] Mascharka, David, and Eric Manley *LIPS: Learning Based Indoor Positioning System Using Mobile Phone-based Sensors, 2016.*

[3] Ramon F. Brena et al. *Evolution of Indoor Positioning Technologies: A Survey, 2017.*