SoccerBeacons: tracking player movements in a soccer game using iBeacons

Suk Hwan Hong
(shong38@gatech.edu)

Motivation

I and my friends from a church gather to play soccer together every week. Although the game is only for a recreational purpose, the game often gets very competitive among players. Players love to compare the performance of each other in a game. Some players are noticeably better than others. However, due to the nature of the group, most players are just average. It is hard to distinguish one player's performance from another, and often "bragging rights" goes to the player who has scored the most in a game. However, using a number of scored goals as a sole metric of a performance of a game is not only unfair but also misleading.

A more appropriate statistics would be measuring "movements" of a player during a game. A movement metric (such as a heat map) during a game can give a better indication about a player's performance (Fig. 1). If we can generate this metric for our weekly games, not only it will generate more entertainments, but also it will bring out weekly soccer game to the next level!

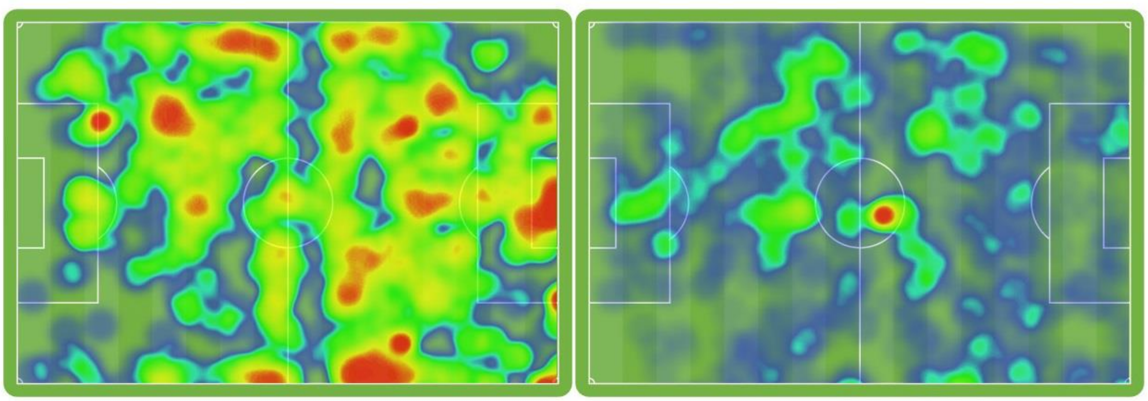

Fig. 1: Heat map from an actual professional soccer game [1]

Overview

In order to measure such metric, we need a sensor that can measure a location of a player with a fine granularity and high frequency. Also, in order to be a practical solution, the sensors need to be cheap and easily available. After ruling out GPS for a possible option due to its coarse granularity and delay, I found a perfect candidate for my application: iBeacon [2].

iBeacons[1] are cheap Bluetooth devices that broadcast signals to its proximity. They have an average accuracy of 0.5 meters to 1 meter and a range of about 70 meters [3]. iBeacons can be purchased from multiple vendors with about $30 each [4]. Tracking

[1] Technically, iBeacon is a protocol standardized by Apple. For convenience, I will call iBeacon-compatible hardware as iBeacon in this proposal.

indoor locations is a common application of iBeacons, and it works best when rooms are rectangular-shaped [5].

Fig. 2: indoor location tracking example [5]

Since a soccer field can be thought as an empty rectangular room, I believe that the same logic that is used to track indoor locations can be applied to a soccer field directly, only at a larger scale.

For this project, I developed an Android Application that can track the movement for any rectangular-shaped field.

Development Environment

Phone: Samsung Galaxy Note 3 with Lollipop (5.0)

Beacons: 9 Kontakt.io Smart Beacons (set to highest signal power and 100ms interval)

IDE: Eclipse Kepler (4.3.2) with ADT (Android Developer Tools) plugin

The source code for Android application is available at:

https://github.com/sukhong/SoccerBeacon

Application Workflow

1. Beacons will be placed in a rectangular-shaped field at certain positions (Fig. 3). Ideally, beacons will be spread out in a field evenly at same interval.

2. After placing all beacons, users manually specify the dimensions of the field and coordinates of each beacon in my Android application.

3. Users have an option to calibrate beacons before starting to track positions. Otherwise, default calibration values will be used. Users may also dynamically calibrate beacons while tracking beacons.

4. Users also have an option to adjust certain parameters to make tracking result more accurate.

5. When a user ready, he/she can start the tracking process. The user carries the phone (with my Android application running) and moves around the field. My Android application will track current position and show a live tracking result on the screen.

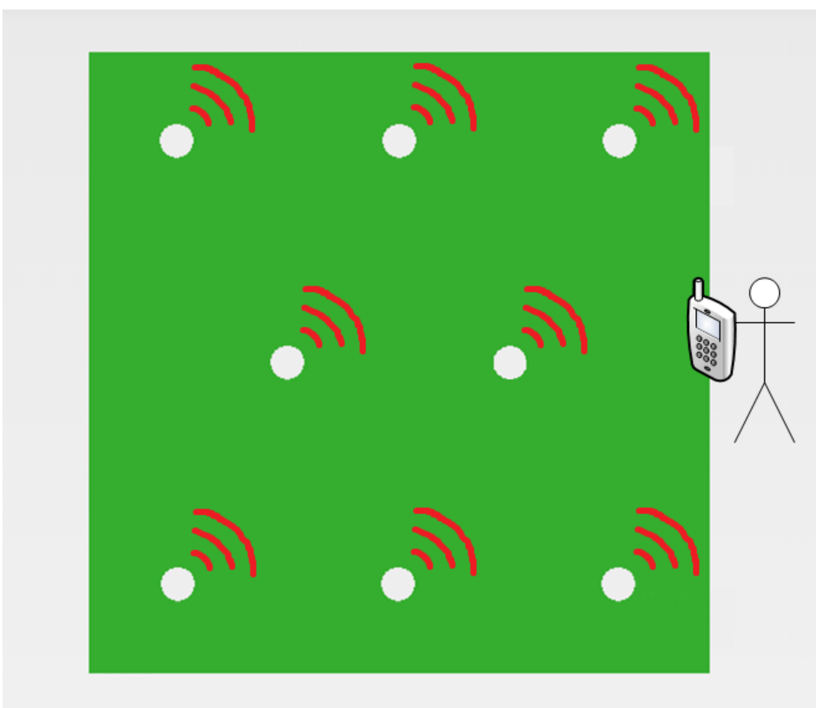

Fig. 3: Field Setup Example

Application User Interface

1. Main Screen (Fig. 4):

This is the landing screen of my Android application. It displays current field setup and shows all appropriate actions that users can take. Users cannot start tracking until field dimensions are set and beacon locations are entered.

2. Field Dimensions Screen (Fig. 5):

This is where you set field dimensions. Users enter field dimensions (width and height) in meters.

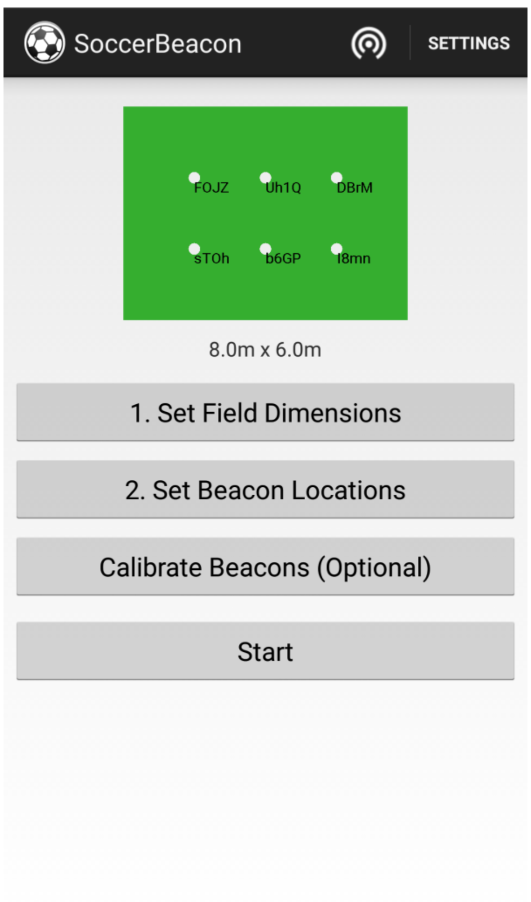

Fig. 4: Main Screen

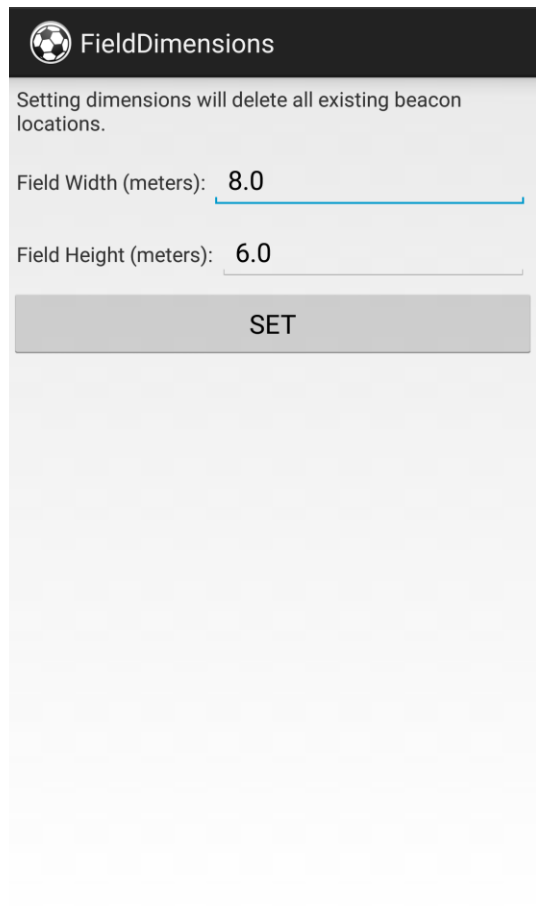

Fig. 5: Field Dimensions Screen

3. Beacon Locations Screen (Fig 6, 7):

This is where you can see the list of beacon locations entered so far, and also enter coordinates for new beacon's location. After you enter coordinates and select "Add Beacon" button, you are directed to a screen where you can select one of nearby beacons.

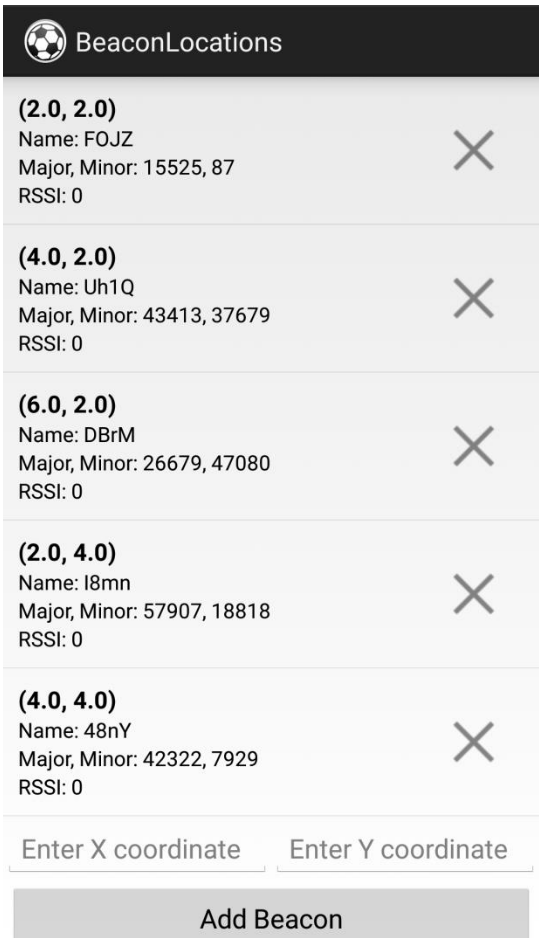

Fig. 6: View/Add beacon Locations

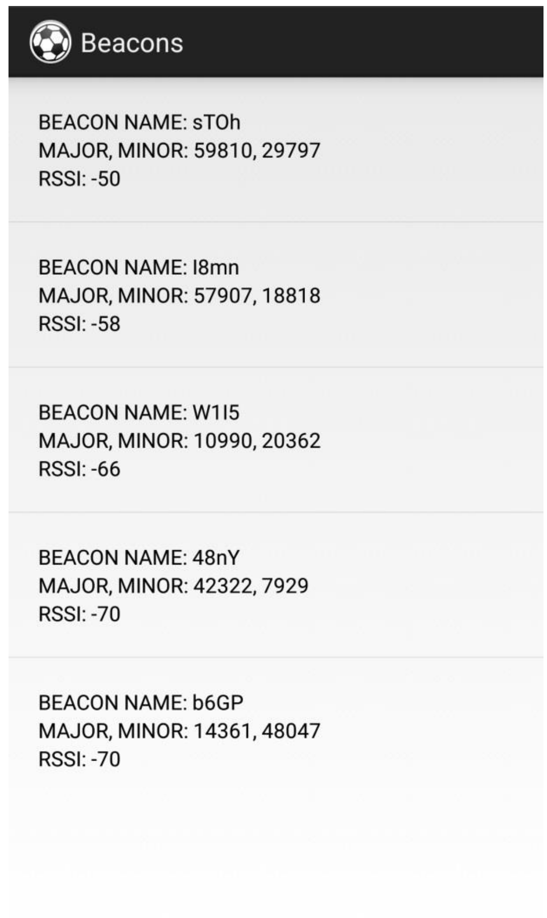

Fig. 7: Select a beacon to add to the field

4. Calibration Screen (Fig 8, 9):

This is where you select a beacon for a manual calibration. Use must stand at 10 different meters from the beacon, and stay still to get the signal power from each location. When user submits the result, application will calculate the calibration parameters and apply it for future distance calculations.

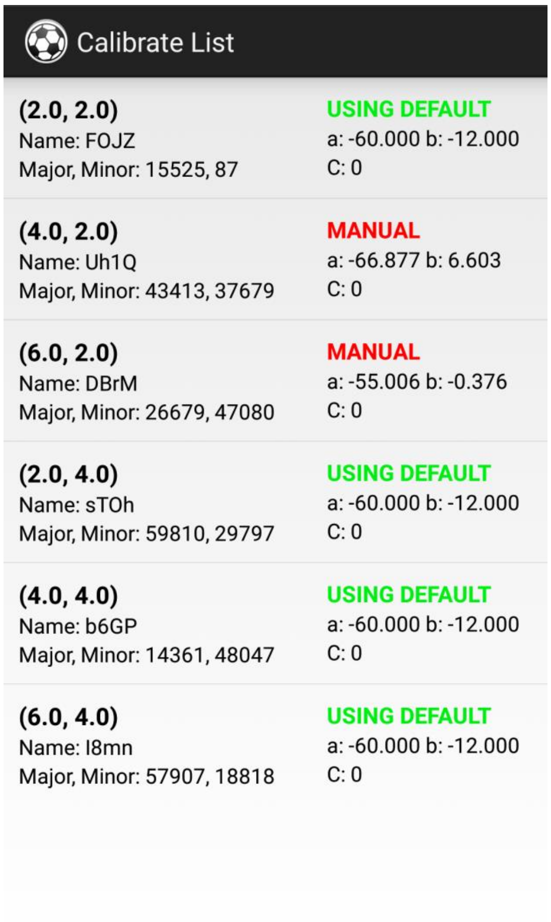

Fig. 8: Select a beacon for calibration

Fig. 9: Manual Calibration

5. Settings Screen (Fig. 10)

This is where you adjust parameters for the application. Each parameter will be discussed later in the report.

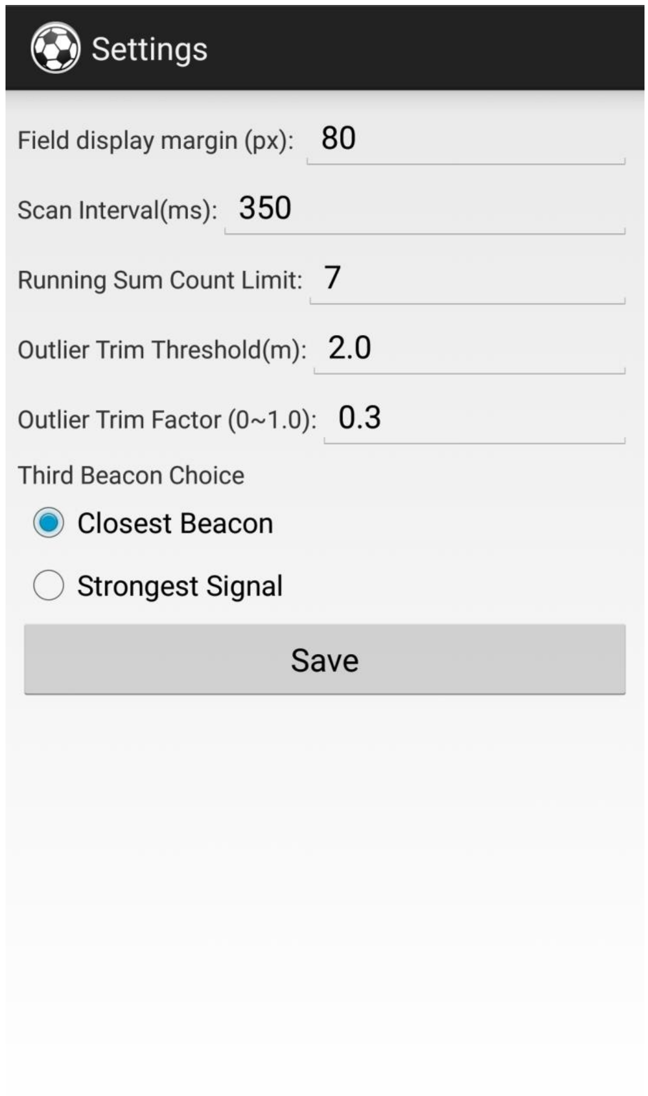

Fig. 10: Settings

6. Track Location Screen (Fig. 11)

This screen displays the live tracking results to the user. Initially, it only displays the current tracked location. When user clicks "START" button on top, it starts to record tracked locations and displays the history of all tracked locations so far. Users can also turn on/off calibration pane on the right side. This panel allows user to apply "dynamic" calibration while tracking is on-going. View Options allow turning on/off beacon range circles and beacon info.

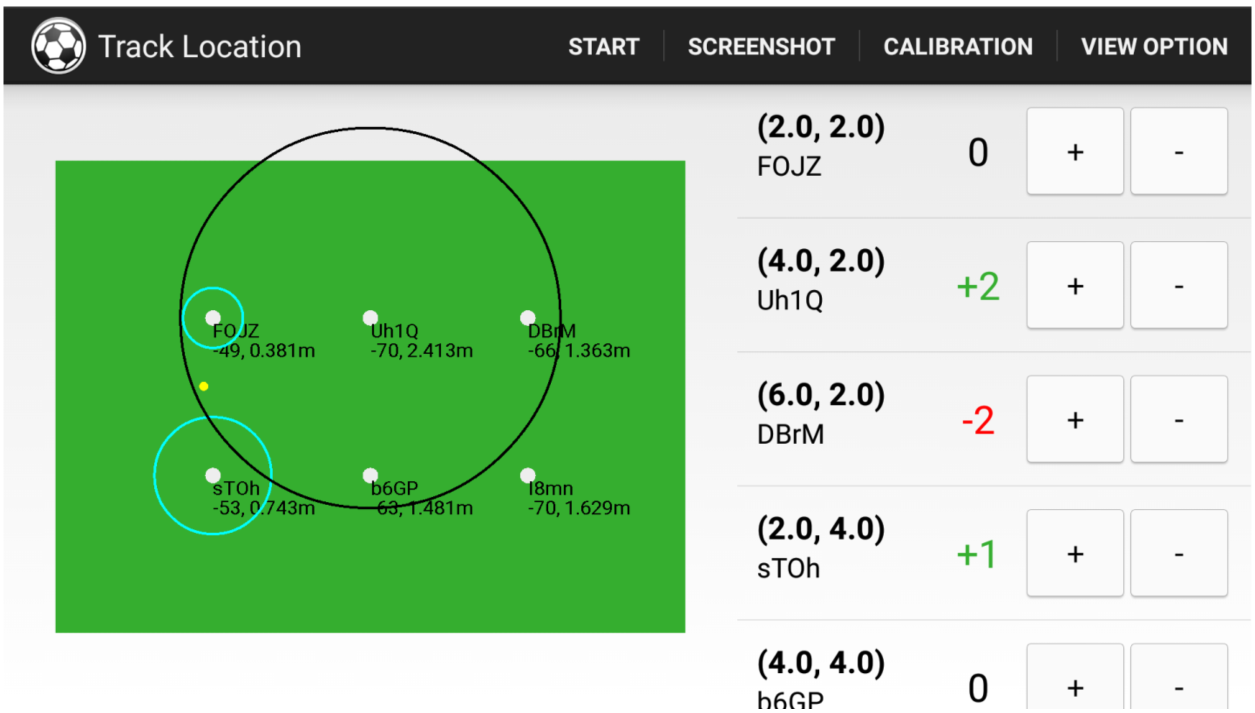

Fig. 11: Track Location Screen

Estimating Distances from measured RSSIs

I had to sample many, many signal data from beacons to get a reasonable dataset to come up with a best-fit formula to estimate the distance.

When we measure the beacon signal power, it comes in the form of Received Signal Strength Indicator (RSSI) in a unit of dbM (decibel-milliwatts).

Since dbM is a logarithmic unit, it makes sense to fit a logarithmic curve y = a * b*ln(x) to the dataset, where y is RSSI, and x is in meters.

I took a sample measured RSSI at 1m to 5m[2] at interval of 1m per each beacon (some beacon measured multiple times), and table 1 shows the result:

	#1	#2	#3	#4	#5	#6	#7
1m	-54.29	-61.381	-59.75	-61.286	-60.455	-60.733	-53.55
2m	-64.226	-67.955	-69.833	-68.375	-69.417	-70.471	-59.167
3m	-67.419	-73.957	-75.056	-72.154	-72.5	-75.308	-63.833
4m	-72.156	-76.818	-77.176	-77.543	-73.833	-76.938	-67.6
5m	-74.839	-79.517	-81.786	-79.583	-76.867	-77.235	-69.036

#8	#9	#10	#11	#12	#13	Avg.
-61.909	-63	-51.32	-55.154	-55.355	-60.733	-58.378
-68.5	-67.909	-57.864	-64.375	-62.677	-70.471	-66.249
-73.417	-68.231	-63.476	-68	-66.677	-75.308	-70.410
-77.214	-76.182	-69.238	-69.4	-68.29	-76.938	-73.794
-80.818	-80	-71.091	-73.19	-70.355	-77.235	-76.273

Table 1: Sample measurements

Figure 12 shows the line that fits the average of sample measurements:

[2] I actually measured data up to 10m (full data available), but there were too much variance in the data above 5m. For current project's purpose, it is sufficient to measure only up to 5m, so I decided to ignore data above 5m. However, we may need to use full data in order to gain better results in full application setting.

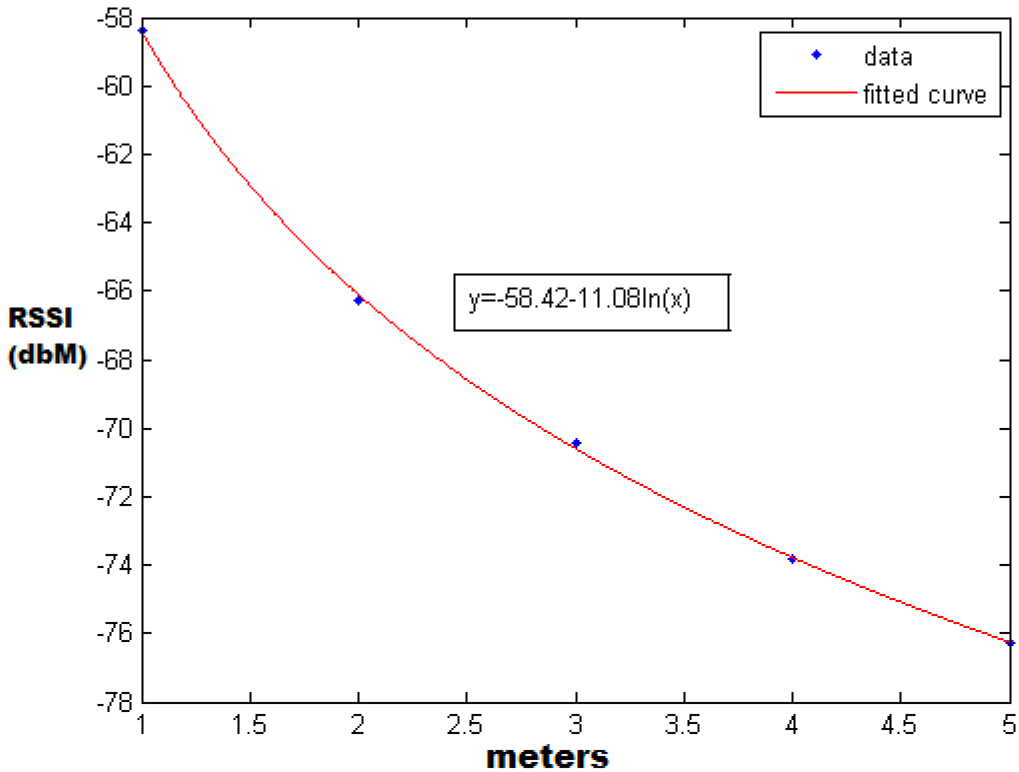

Fig. 12: Fitted curve across average sampled data

Using this line, we can estimate the distance based on the measured RSSI. This line's coefficient and the intercept (a, b) became the default calibration parameters we used for the application.

However, there may be cases where you need to calibrate further. Therefore, I provided an option to manually calibrate the beacon (completely new line). Also, I provided an option to dynamically adjust the default fitted curve while tracking the location.

In the latter option, users can adjust the third parameter c (initially 0), so that the curve shifts up and down: $y = a * b*\ln(x) + c$

A beacon gets "weaker" as c increases. A beacon gets "stronger" as c decreases.

Algorithms for determining the location

I used some variation of trilateration algorithm (used by GPS) to determine the location given distances from each beacon.

In order to determine the location, we need distances from at least three beacons.

My algorithm works as:

1. Find two beacons that are giving the strongest signals. Given two "range circles" from each beacon, there are three cases:

    a. two range circles intersect with each other: take the middle point
    b. one circle is contained in another
    c. two circles are disjoint

In each case, I use a (not so) simple geometry to find a "base" point that will be used for later algorithms.

Figure 13 describes these three cases:

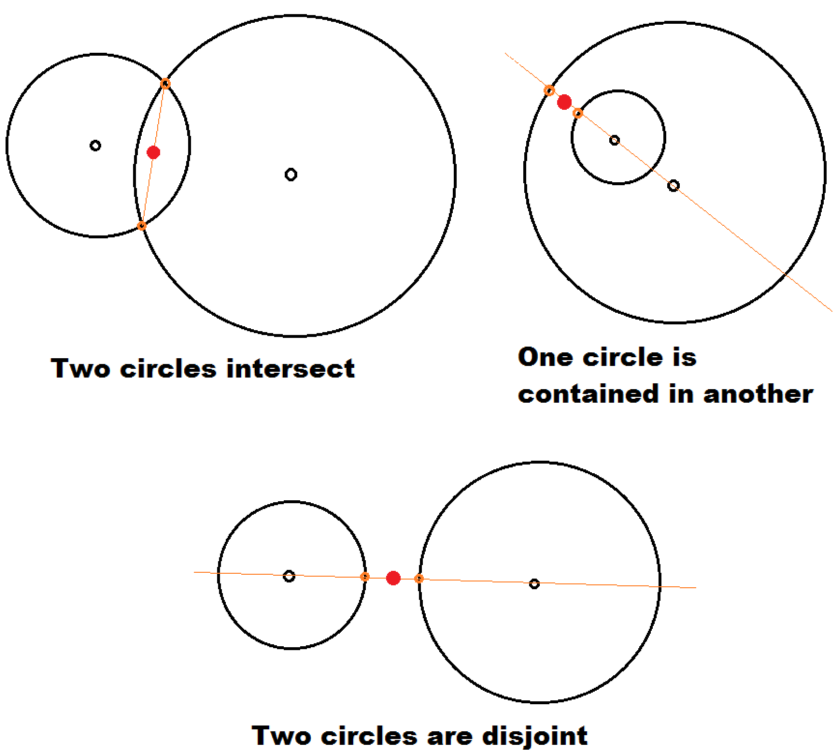

Fig. 13: three cases

2. Then using from this base point, I find a third beacon to use to determine the location. There are two possible choices here:

   a. Use the closest beacon from the base point

   b. Use the beacon that is giving the strongest signal

The rationale behind choosing closest beacon is that there is a high probability that the closest beacon from base point is a correct choice even if it doesn't give the strongest signal.

Figure 14 describes this situation, and resulting position can be different significantly.

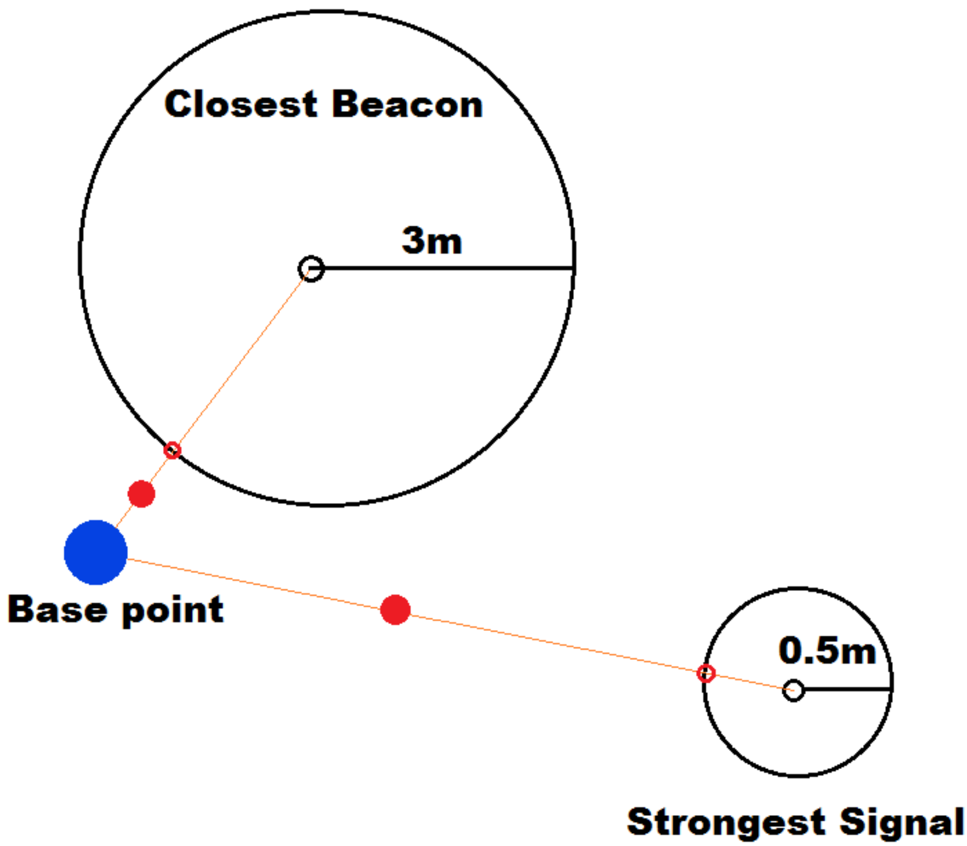

Fig. 14: Choosing third beacon

Using either choice, we can now determine the location by getting the average position between base point and the closest point that lies on the range circle of third beacon.

We can use more than 3 beacons to determine a location, but empirically, 3 beacons worked the best.

Configurable Parameters

Beacon signals fluctuated from time to time, and the noise was inevitable during the measurements.

Therefore, I came up with some parameters that could help reducing the noise in signal measurements and improve the results.

Also, I introduced some useful configurable parameters that can be adjusted to fit into different situations.

In total, 6 parameters are configurable by users.

1. **Field display margin**: This is a margin in pixels around the field in the display during tracking process. A large margin helps users to detect outlier positions outside of the field.

2. **Scan Interval**: This is an interval in milliseconds between each scan of beacon signals. Fast scan interval gives you finer granularity in each position. However, fast scan interval has some chance of missing some beacon completely. Low scan interval gives you lower granularity, but it is more solid in a sense that it will most likely not miss any beacon's signal.

3. **Running Sum Count Limit**: When we measure RSSI from a beacon, instead of measuring a single instance, we measure the running sum of RSSI from a beacon. This number tells you how many previous RSSIs to use for calculating running sum. Higher running sum count limit helps to reduce the noise in signal measurements.

4. **Outlier Trim threshold**: This is the limit in meters to apply outlier trim factor discussed below.

5. **Outlier trim factor**: This is a number between 0~1 that is multiplied to any distance above outlier trim threshold. When a large distance is suddenly discovered, you don't want to move to it right away, and there is a high probability that this large distance is an outlier. Instead, you apply this trim factor so that you reduce the noise.

6. **Third Beacon Choice**: Pick an algorithm used for choosing a third beacon in determining the location. Two choices were discussed already in Algorithms section above.

In Experiments section, I run an experiment with varying degree of each parameter to show how effective each parameter is.

Experiments

For experiments, I set up and used 10m x 10m square field in Klaus Atrium (Fig. 15). 8 beacons were placed evenly across the field[3]. The horizontal distance between two beacons was 3m, and the margin around beacons was 2m. When I moved across the field, I used two movement patterns when applicable: move along the border and move diagonally across the field.

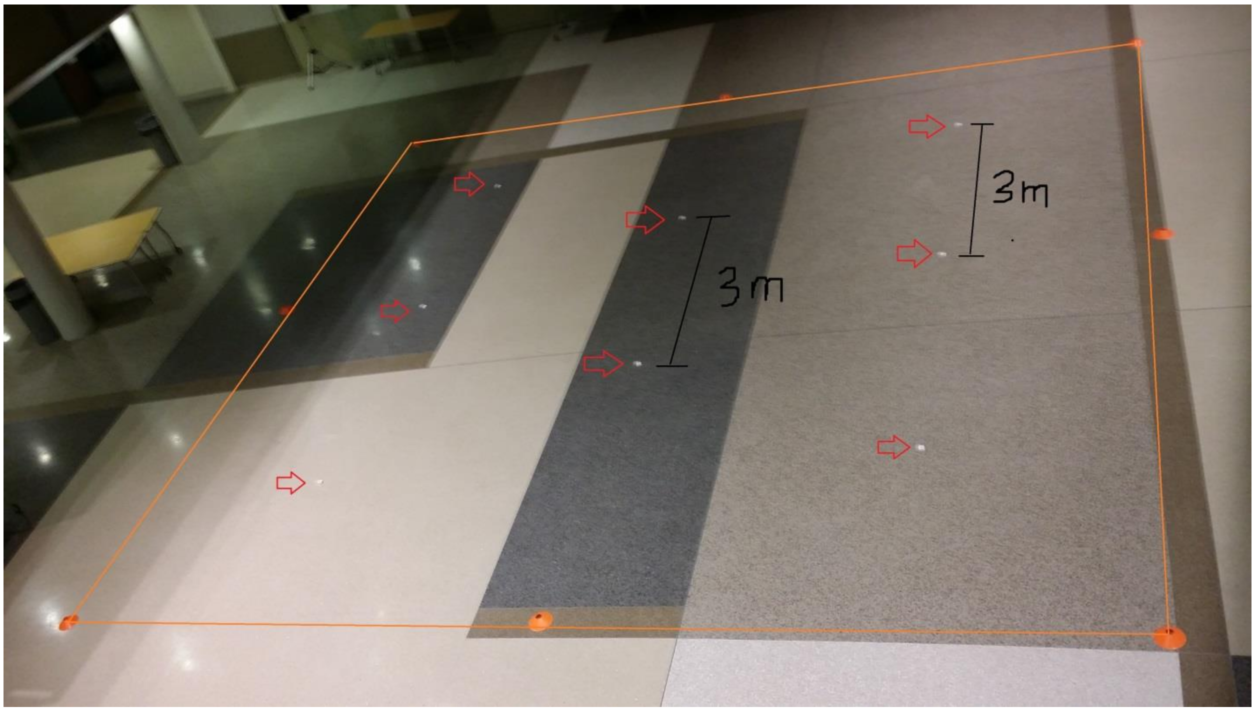

Fig. 15: Experiment Setup

I believe that if these two movement patterns are recognized well by my application, it can recognize all movement patterns as well.

When some parameters are not specified, following default values are used for them:

350ms scan interval, 7 Running Sum Count, 2.0 outlier threshold with 0.3 trim factor, and closest third beacon

Here are experiments results:

[3] Initially, I had 9 beacons, but 1 beacon suddenly died. I believe that 9 beacons would have performed much better, although 8 beacons still performed well.

Third Beacon Choice Test (Border and Diagonal)

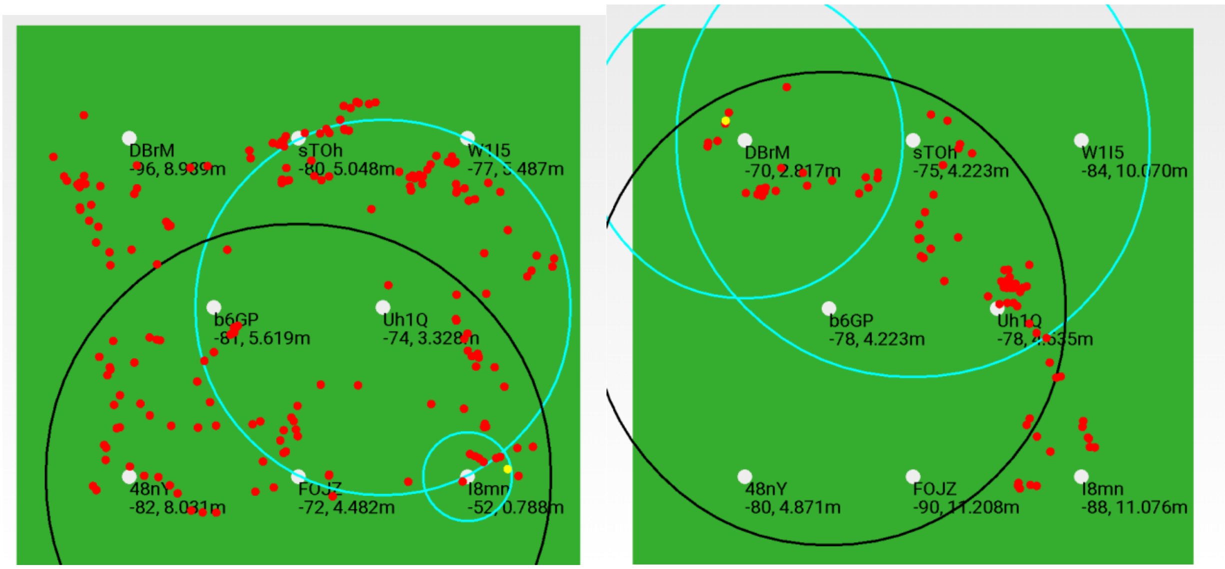

Fig. 16: Strongest Signal (border and diagonal)

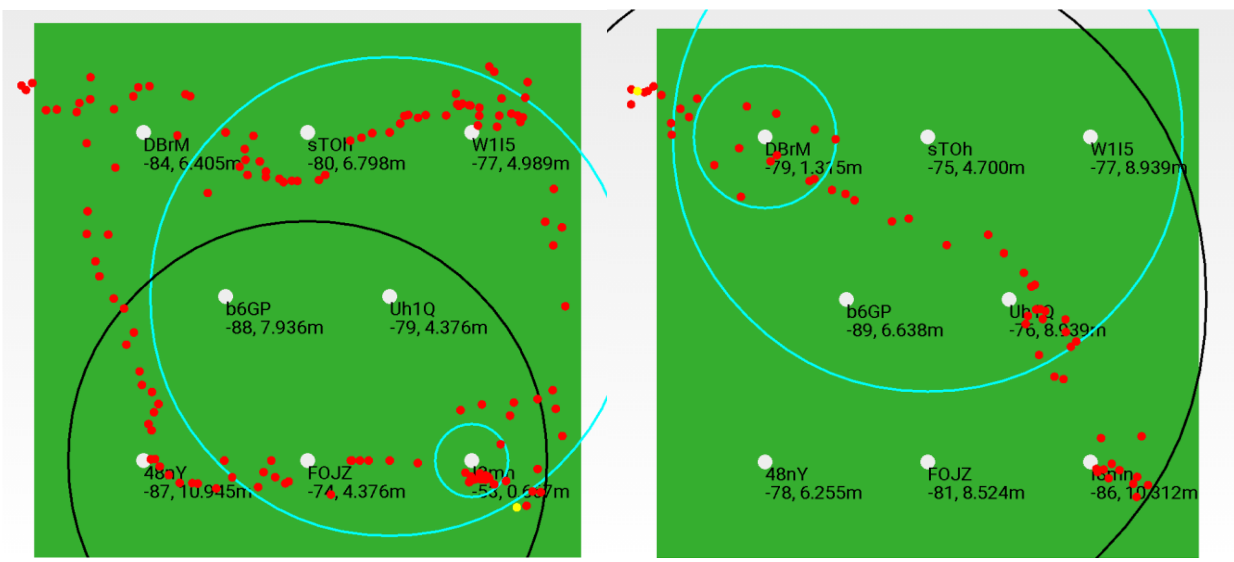

Fig. 17: Closest Beacon (border and diagonal)

Closest Beacon method gives much better result than strongest signal method. I believe that this is due to the fact that signals are not really accurate in further distance, and therefore, there is a high probability of strongest signal not being correct choice for third beacon. There is a higher probability that closest beacon from first two highest signal beacons are the correct choice.

Scan Interval Test (Border Only)

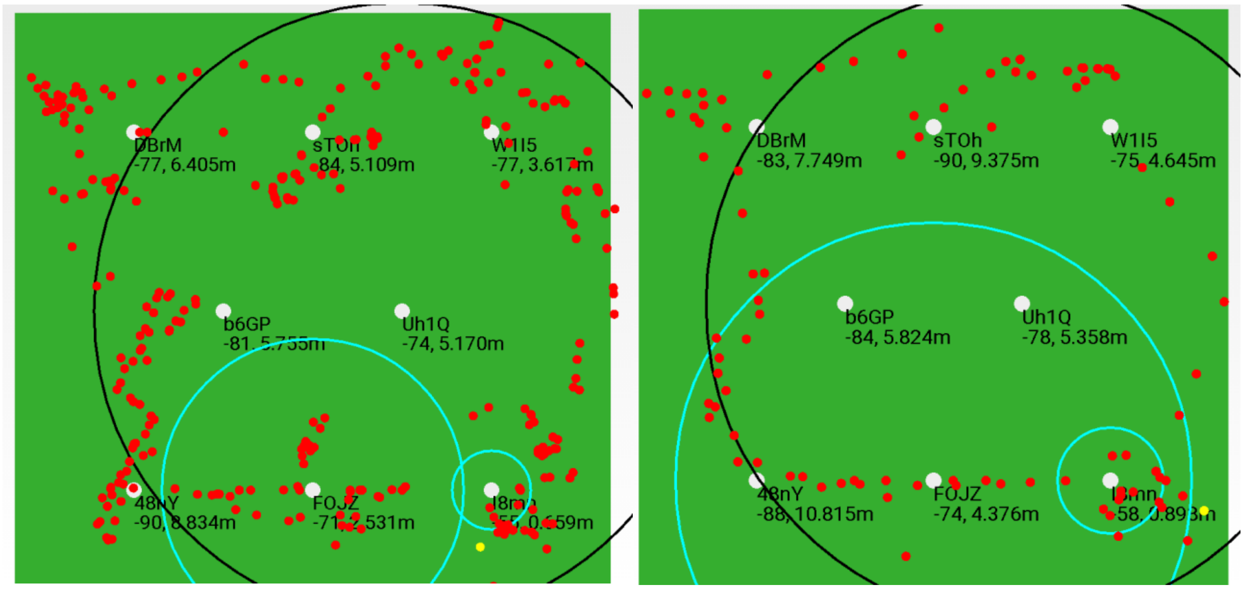

Fig. 18: 150ms (Left) and 350ms (Right)

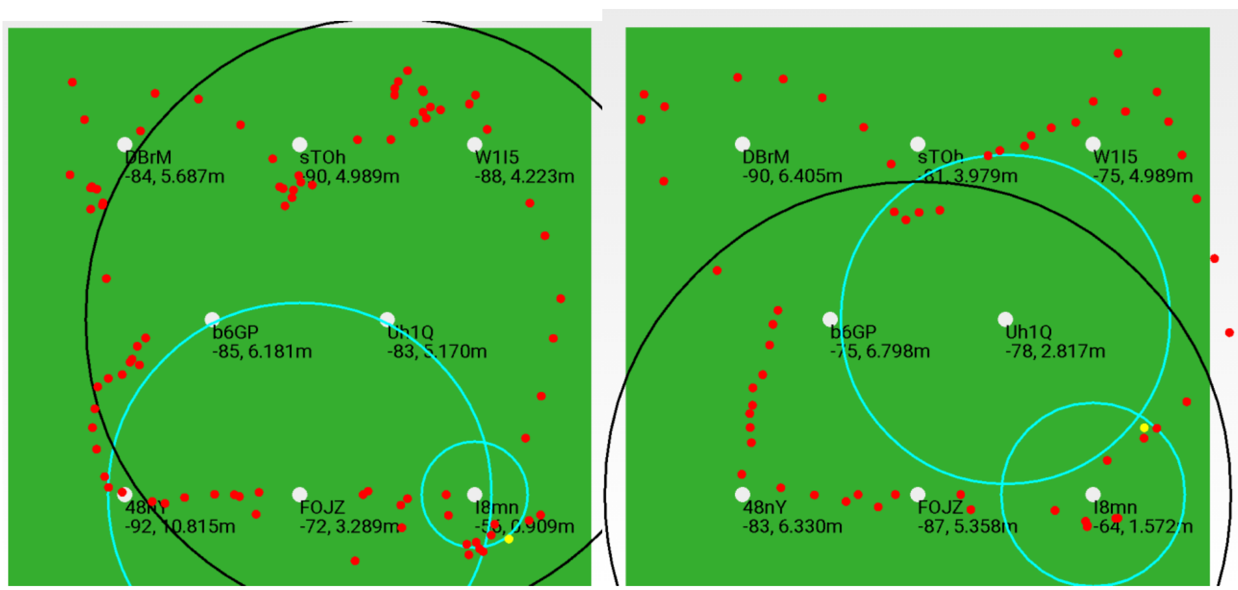

Fig. 19: 700ms (Left) and 1000ms (Right)

Faster scan interval gives much finer granularity, but at the same time, it is possible to miss some beacon's data, so there may be some noise introduced as well.

Compared to 150ms, 1000ms interval gives result that may be too coarse for certain applications.

Running Sum Count Limit Test (Border Only)

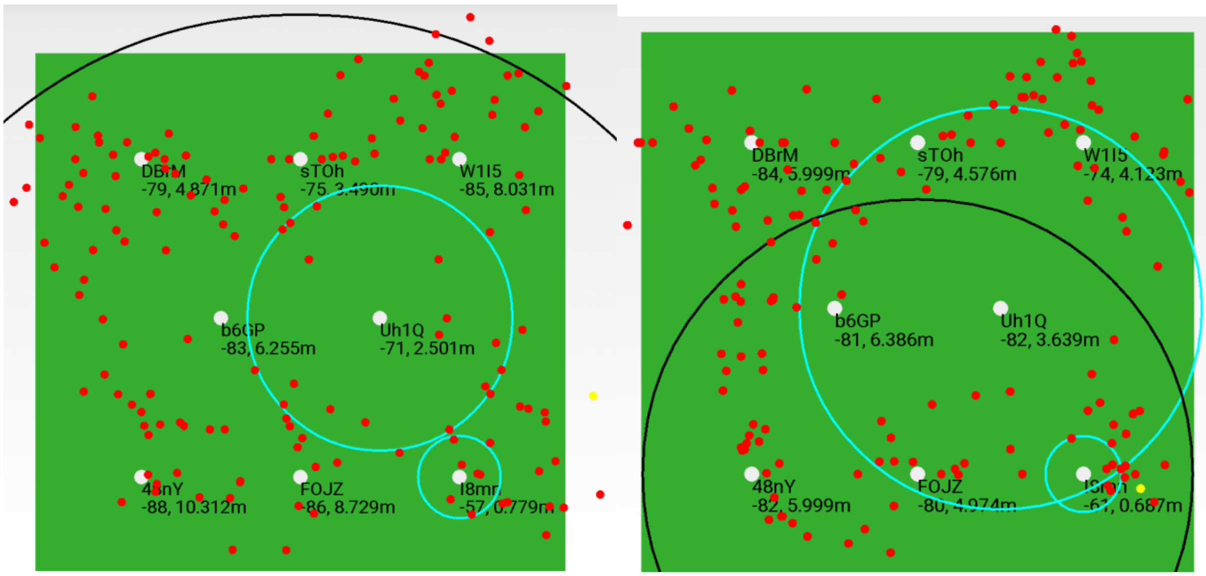

Fig. 20: 1 Running Sum Count (Left) and 4 Running Sum Count (Right)

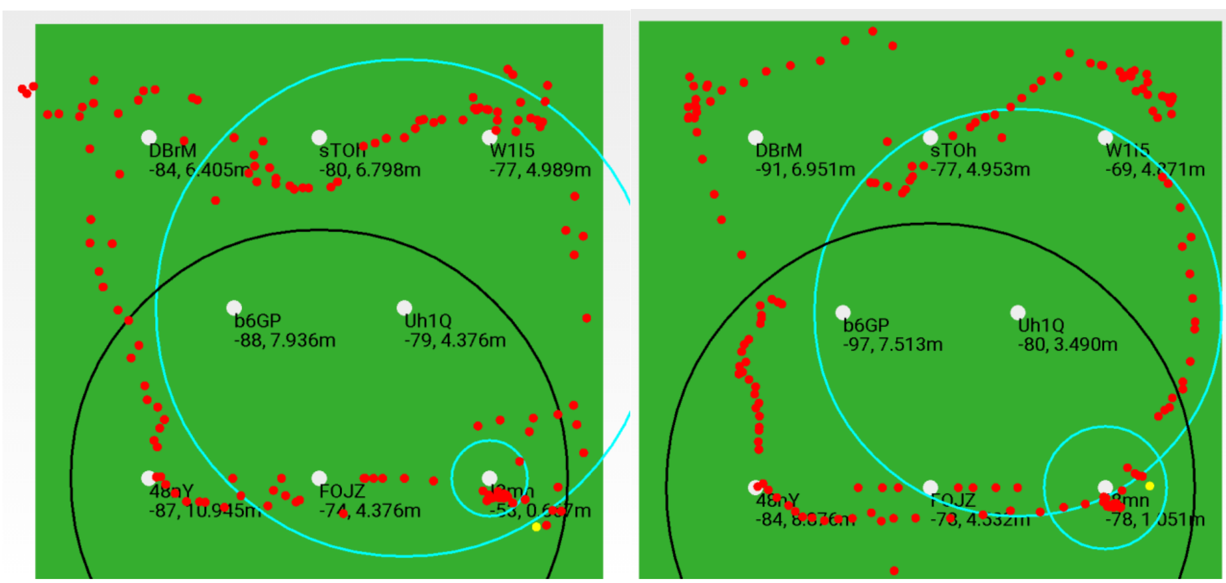

Fig. 21: 7 Running Sum Count (Left) and 15 Running Sum Count (Right)

Running sum averaging received signals have a significant impact on the result. As you can see, with no running sum average, the data are too scattered to be useful. However, making running sum count too large may also introduce a problem where dynamic movements are not tracked immediately.

Outlier Trim Factor Test (Border Only)

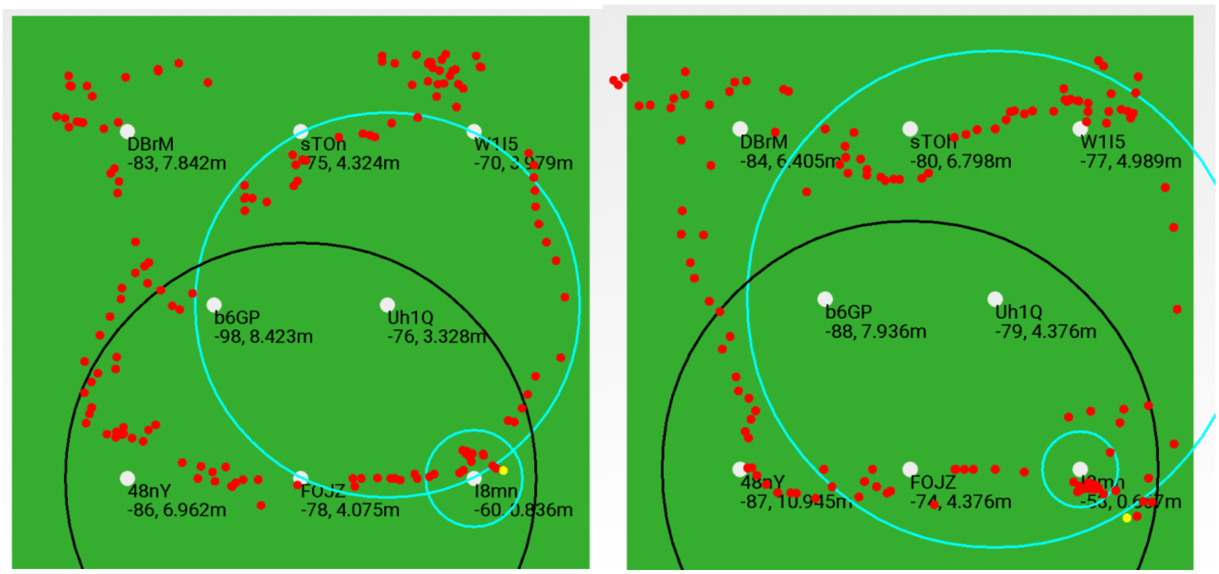

Fig. 22: 7 Trim Factor 0.1 (Left) and Trim Factor 0.3 (Right)

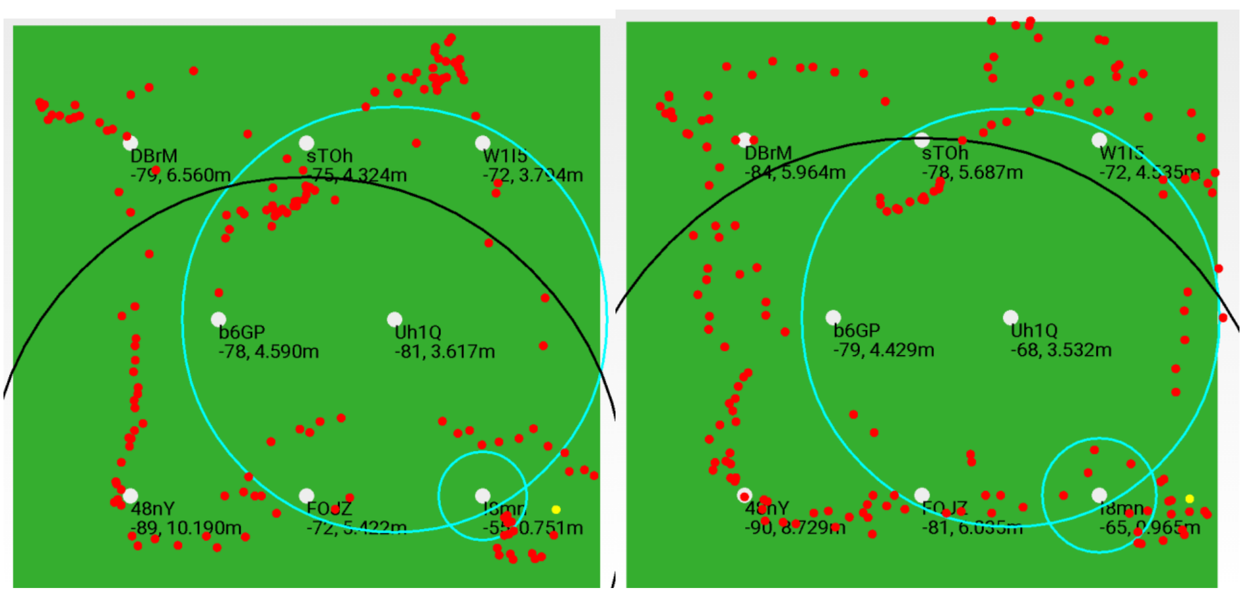

Fig. 23: 7 Trim Factor 0.7 (Left) and Trim Factor 1.0 (Right)

Contrary to my belief, outlier trim factor had almost no visible effects. I believe that running sum average is already doing the same work that outlier trim factor is supposed to do, so it is duplicate to have this parameter along with running sum count parameter.

3m vs. 5m horizontal distance between beacons

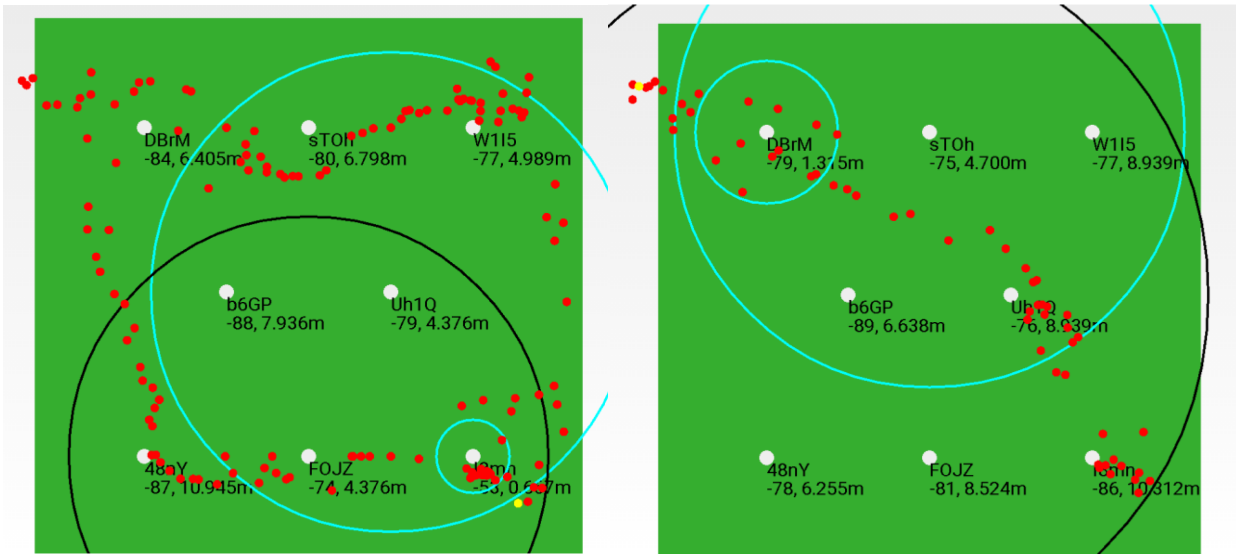

Fig. 24: 3m horizontal distance between beacons

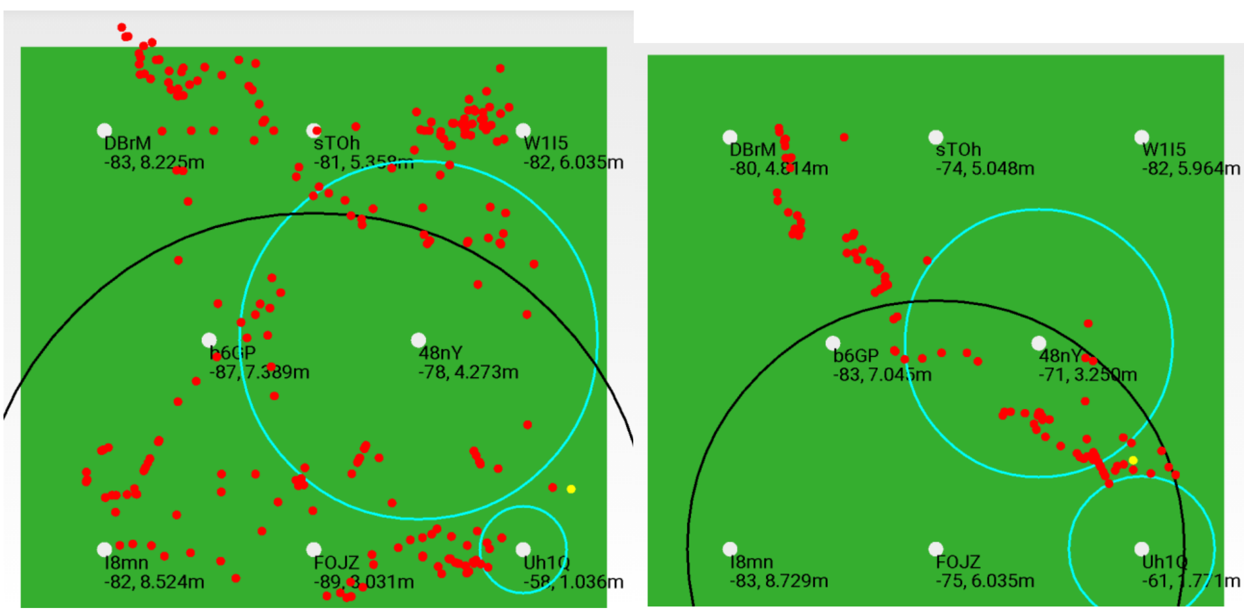

Fig. 25: 5m horizontal distance between beacons

In 5m, the result is definitely worse than 3m version. However, I attribute this to the fact that I didn't have a middle beacon along vertical edges of the field (because my 9th beacon died, I had to use this scheme…). I believe that with 9th beacon added, the result will be better.

For diagonal movements, both distances showed similar results.

Lessons Learned

I learned that signal strengths from iBeacons can vary widely from time to time, so that the noise reduction and a smart algorithm to "predict" the right movement is the key to the success of tracking the movement correctly.

Also, below certain signal strength, it is impossible to get an accurate estimate of distances from a beacon. Therefore, it is critical to rely on first few beacons that have the strongest signals to you, because there is a high probability that those beacons are giving correct signals to you.

I also learned that the RELATIVE distances between the beacons are far more important than ABSOLUSTE distances from the beacons. Therefore, a smart algorithm to predict the location by using the relative distances between the beacons is a key to success.

Using appropriate third beacon choice and incorporating various noise-filtering methods, I was able to track the movement of a person successfully in a 10m x 10m environment, up to 14m x 14m environment.

Finally, I learned that I overestimated the capability of beacons before the project. Some articles mentioned that iBeacons can be used to measure the distance up to 10m, but I found that claim far too optimistic. From my empirical experience, I found that any distance above 5m conveyed almost no finer granularity than "I am far."

Applying it to a soccer field…?

Due to the lessons learned during the project, I had to abort the ambition to applying it to the soccer field.

The main reason is the cost.

My experiments showed that my application works marginally at 5m horizontal distance between beacons. Typical soccer field's dimension is 100m x 60m.

Let's say that we apply to a much smaller soccer field, say 70m x 40m.

Even with this smaller field, I need at least 13*7 = 91 beacons (and 209 beacons for a regular soccer field) to cover the whole field!

That will cost me more than $1,000 dollars even with the cheapest beacons.

However, apart from the cost, this project clearly showed that the beacons can be used to track outdoor movements.

Therefore, for a question of whether it is “feasible” to use iBeacons to track movements of a soccer player in a regular soccer field, I will say **YES**.

But whether it is “practical” to apply this project to the soccer field, I have to say **NOT YET.**

I hope that the iBeacons improve in the future so that it also becomes “practical” to apply this project to a soccer field.

Future Works

1. Automatic calibration of beacons
2. Better algorithm for determining the location given measured beacon signals
3. Ability to save and replay tracking data
4. Ability to stream tracking information to a separate device or a server

References

[1] Brazil vs. Germany semi-finals match heat map in 2014 World Cup

[2] iBeacon vs NFC vs GPS: Which Indoor Location Technology will your Business Benefit from? - http://blog.beaconstac.com/2015/07/ibeacon-vs-nfc-vs-gps-which-indoor-location-technology-will-your-business-benefit-from/

[3] iBeacon Wikipedia - https://en.wikipedia.org/wiki/IBeacon

[4] Kontakt.io Store - https://store.kontakt.io/our-products/1-ibeacon-smart-beacon.html

[5] Estimote Indoor Location - http://estimote.com/indoor/