# Beacon-based Indoor Localization by using Android Device

ZHANG Lin, HUANG ZiJian

Supervisor: S.H. Gary Chan

UROP 1100 Spring 2018

The Hong Kong University of Science and Technology

## Abstract

abstract---Indoor localization has been a hot topic in recent years, leading many researchers to develop different kinds of localization methods using different sensors. This report will mainly discuss our implement of indoor localization using beacons and Android devices. We will illustrate how we choose our hardware devices and software platforms, and the structure of our system, as well as our mathematical foundations. The system we implemented is not so complex, we only implemented the skeleton structure of the system, but the function is completed and can be used for further developing. The structure basically is a server-client system structure and will track the users and save the users' location into our database. Further improvement will include using more advanced localization algorithm, adding more functions to our front-end page, etc.

## I. INTRODUCTION

The purpose of this report is to explain the way to retrieve user location data by using the Bluetooth function of mobile phone and iBeacon which can send signal to mobile phone. The report will also recommend one better way of indoors localization in some simple assumptions.

About 55 years ago, GPS was invented and people can locate things on the earth everywhere outdoors. However, because the special space properties indoor, the indoor localization problem is in tardy progress and is still not solved thoroughly. Nowadays, there are many ways to localization indoor, such as Triangulation, Fingerprinting, Dead-reckoning and so on. However, we want to find a better way to locate things indoors or improve one existing method.

Two students involved in the real testing, in which one held the mobile phone and got the signal intensity in a fixed position while the other moved the iBeacon from the closer to the farther position from the mobile phone. Function fitting was involved when trying to find the relationship between the signal intensity and the distance between the mobile phone and iBeacon. References were also made to books and articles.

This research is just a simple model which only uses two iBeacons and the easy formula in triangle which is not much concerned about the random error and error detecting.

## II. system construction

In this section, we will explain in detail how we built the whole indoor localization system in both hardware and software selection and construction.

### *A. hardware devices*

In this project, our hardware devices mainly comprise beacon and Android device. Beacon is widely used nowadays in building localization systems for its low price, low energy consumption and portability. The strength of signal broadcast by beacon can be detected by many wireless devices. Android device is the most frequently used wireless device and Android program can be transplanted into different kinds of devices, which makes it suitable enough for this project. However, both devices still have defects in localization, and we will analyze them at the end of either section.

### a. Beacon

Unlike other heavy Bluetooth devices, beacon is small and portable. In this way beacon can be placed in any places such as construction site, university, shopping mall, etc. And it is also invaluable, which can reduce the cost a lot and make localization technique more practical and popular.

In this project, we use MiniBeacon produced by Minew Technologies. It does not support programming but has 8 adjustable levels of transmission power, from -30dBm to +4dBm, which can be easily adjusted through an app. This is satisfying enough for our project. By setting the transmission power to level 8, the signal of beacon can be received in a distance 50 meter far away.

However, detecting signal strength only using beacons have unneglectable flaws. The signal sent from beacon is quite unstable and could have very large errors. In the future, when we need to receive more accurate signal to achieve higher accuracy in localization, using beacon only is not recommended. We have many other hardware devices such as magnetic and WIFI devices. Using the combination of different devices is also achievable. But for this UROP project, using beacons is just appropriate enough.

### b. Android Device

Along with the popularization of smartphones is the popularization of Android device. Android device can have different prices ranging from several hundred to infinity HKD, therefore we can choose the relatively inexpensive ones for experimental use. In this project we use Android smartphones and electronic watches. Besides, the most important reason why we use Android device is that we have already had many Android APIs for functions like Bluetooth signal detection and Google map API. This can save us a lot of trouble when building the software system.

However, Android device also has flaws. Its speed of scanning Bluetooth signal can be very slow, sometimes even several seconds per scan. This result is unsatisfying when we are detecting a moving object, which forces us to reduce the update frequency. And if we also consider the errors existed in Bluetooth signal, the distances we get may be too different from the real ones. Fortunately, using multiple beacon- based localization techniques and more advanced algorithm can solve the problem, but that's too complex for our project.

### *B. Software Systems*

In this section, we will show our software system structure. In particular, our system consists of three parts: a server, an online webpage and an Android app. We will explain how each of them work individually by introducing what techniques we have used and how they function. Then we will see how the organic system works by using a flowchart. In each individual part, we will also emphasize the important concepts by reviewing those troubles we have met.

#### a. server platform

In computing, a server is a computer program or a device that provides functionality for other programs or devices, called "clients". In this project, we need a server with an embedded database to store the location information of all the users and beacons. We may need to add, delete or modify user data in real time, keeping updating the database. Therefore, the server is the core part of our project. However, building a server from scratch is very difficult since the inner structure of server is very complex. So, we need to choose some tools to assist us.

There are many existed packages for us to design a server in a convenient way. Among all of them we choose to use Django. Django is written in python, easily to use. The inner structure is built well, and we only need to modify several files to achieve our own server functions. In addition, Django has complete online tutorial sections, which can reduce us a lot of work in learning it. Django also has a database inside and has already written some webpages for us to modify the data in the database in a user-friendly way, for example, the administrator webpage.

### b. Webpage

The webpages that Django can provide is limited, and far from satisfying our need. We need to have a local map, on which the locations of all the users and beacons are labelled and kept updating. Basically our webpage is just a front-end and client. To achieve this function, we need to write our own html and JavaScript file and add it into Django. We use Google map API of JavaScript and HTTP Request in this project.

Google map supports a number of mouse event functions including click, zoom in and zoom out, rotate. And the map can be shown in different styles: map mode with or without terrain and satellite mode with or without labels. These all functions make it very user-friendly and attractive for our webpage, saving us a lot of time for polishing the front-end. Besides, Google map API also has a great bunch of functions for us to achieve different effects like adding labels. In this project, we need to label different users and beacons, so Google map is an appropriate choice.

Our webpage should also act like a client to send request to server in order to retrieve the location data, label and update them in real-time. This requires sending GET and POST request to server. GET request is for retrieving data from server's database, it will not change the database. POST request is for sending real-time data to database to modify data. For webpage, it only requires retrieving data, so we only use GET request. The use of POST request will be illustrated in Android development section after.

During our development procedure, we met a lot of problems because of our unfamiliarity to Django. For Django, sharing miscellaneous files like images with html file bothered us a lot. Then we found miscellaneous files should be stored as static file in a static folder in Django.

### c. Android App

Our user location information is sent from individual online users. So our users should have a device to detect the beacon signal and send their location data to central server. At the same time the local device should also act like the webpage to show all current users location to allow all users communicate with each other. From the perspective of software, Android has provided its own API to allow us create an App easily. And

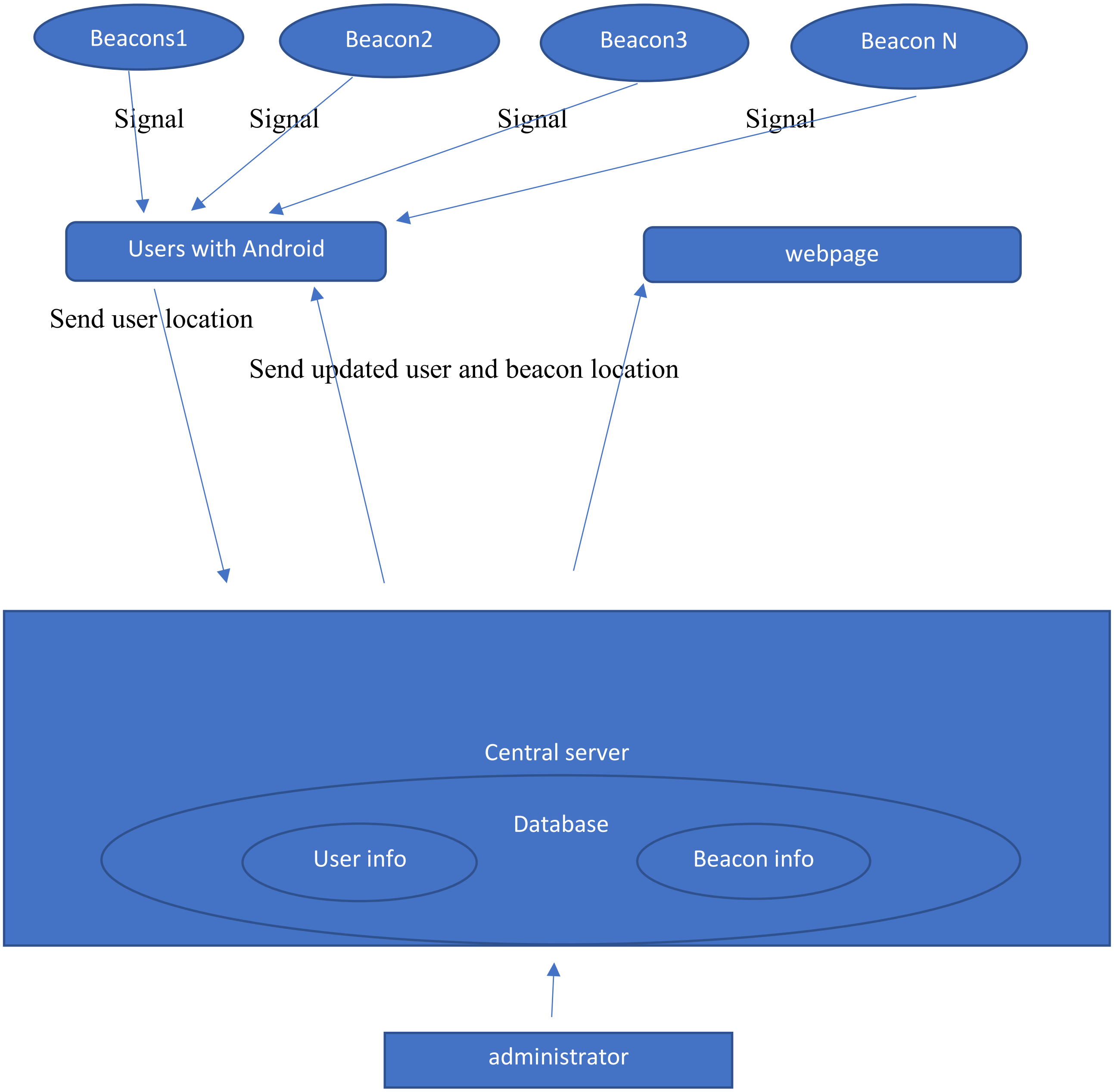
Google map also has Android API for us to use in the same way as the JavaScript API we have mentioned in section b.

For Android API, our focus is to scan beacons by using Bluetooth and calculate the user location through the received RSSI signal strength. The calculation of location is about algorithm, which will be illustrated in part III mathematical foundations. For Google map API, we use it in the same way as JavaScript Google map API.

The difficult technical trouble is mostly about multiple threads processing. Our main and attractive function is to update data in real-time, which requires us to add a new thread other than the main thread. But the data sharing is not easily done as html is. In the new thread, we need to scan beacon device and update the location data in the main thread, and most importantly, sending Http request to server. But sending Http request can only be done in main thread, which requires adding an extra main thread processing function inside the new thread. The difficult part also lied in the variable processing since the variable updated in the new thread may not be updated in the main thread. In order to solve this, we need to arrange the order of new thread and main thread in an appropriate way.

### d. Combination of above Individuals Systems

Basically, the whole system achieves the function to locate multiple individuals through a set of beacons in a local area. The administrators place the preplaced beacons, the latitude and longitude of which are recoded, in a certain region. By using Android devices, individuals in the local area detect beacon signals, calculate their locations through the preplaced beacons, and send request to central server to update their user locations. The server will also send users' updated locations back to Android devices, on which individuals can check their and beacons' position at any time. The users' and bacons' location can also be checked on our webpages at any time if you are authorized. Our webpage will act like a client to retrieve data from central server and show locations on the map. The central database can be modified by administrators at any time, including managing user's location. Below is figure to show the whole procedure.



## III. Mathematical foundations

In this section we will introduce the mathematical knowledge and model on which our project is based. The mathematical part is all about how to transfer the data we have collected into distance in meters. In particular, we collect RSSI signals from different beacons and use normal regression method to calculate the real distance based on our mathematical model. In the following part, we will demonstrate them individually.

### *A. Mathematical Model*

Our mathematical model is based on the most popular RSSI-Distance function:

$$\mathrm{RSSI} = \mathrm{TxPower} - 10*\mathrm{Gamma}*\lg(\mathrm{distance})$$

TxPower: transmission power of beacon

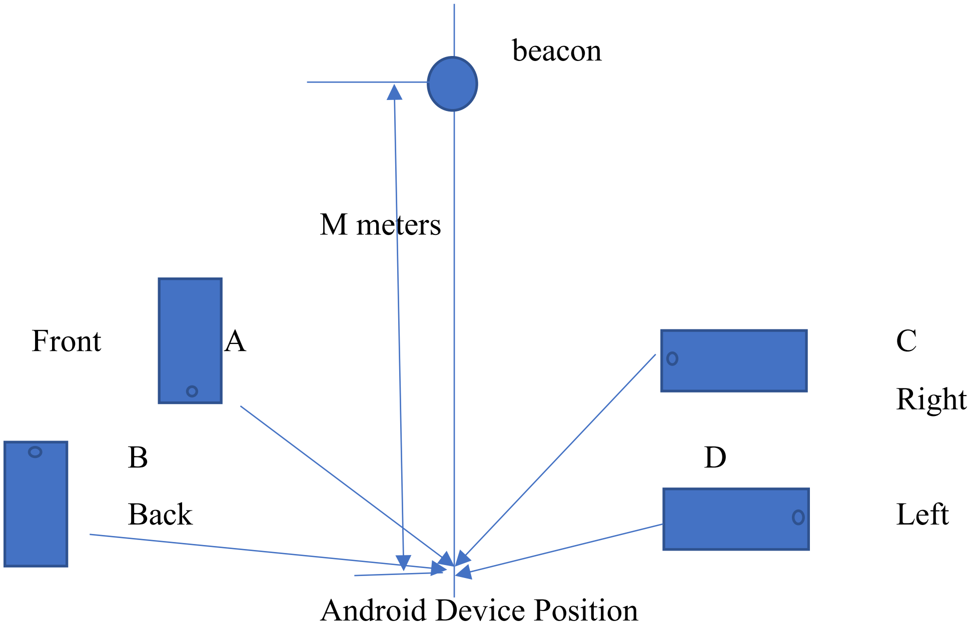
Gamma: coefficient

Distance: real distance from beacon to user / meter

In this equation, we can see that the distance and RSSI have an exponential order relation rather than linear relation. And the TxPower and Gamma value can not be found directly. Therefore, we need to have a set of training values of different distance and corresponding RSSI, then use regression to find the coefficients.

### *B. Experiment Method*

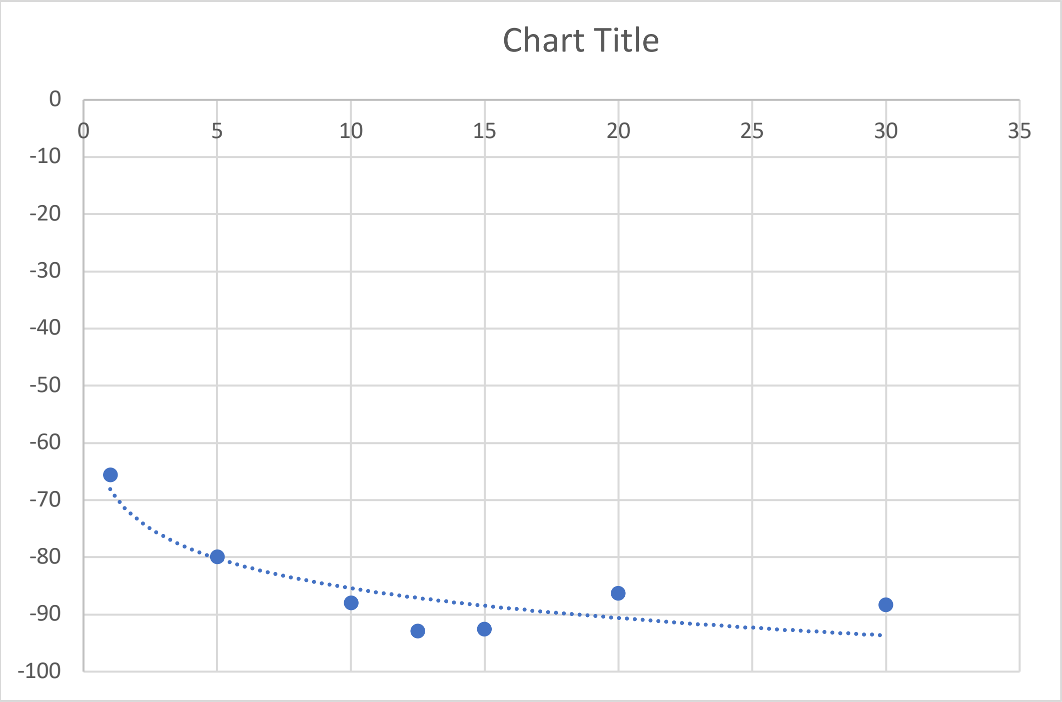
In order to get different real values of distance and RSSI, we need to use an Android device to test it. Our experiment method is getting different RSSI values according to different distance. And in order to have a more accurate result, we place our Android device in four different directions and choose the average value as the final data.



Final RSSI value of distance M is (A+B+C+D)/4

### *C. Regression*

Regression from scratch is difficult to handle, therefore we use EXCEL in Microsoft Company to help us find the real curve between RSSI and distance. Here is our result:



After regression, we can settle those values. Our final result is:

$$RSSI = -63.464 - 7.415\ln(\text{distance})$$

## IV. Conclusion and potential improvement

In conclusion, this iBeacon method which use only two iBeacons is the basis for all other methods using iBeacon because in this model, we ignored some random errors and two iBeacons is the minimum number of iBeacon that we can use to get the third point's position with the known three lines' length.

There are some potential improvements. First of all, we can let the difference between two different length smaller so that we can get a more precise logarithm function of the relation between signal intensity received by mobile phone and the distance between mobile phone and iBeacon. Secondly, we can use more iBeacons to locate a user because the random error will becomes smaller when there are more data to refer to. The third potential improvement, which is also the most important, is that we can use some better algorithms to locate a point, which we will do it later.