Indoor Localization and Fingerprinting across Multiple Smartphones

Internship Report

MEHDI Nima
Supervisor: Bart Van Vliet
Tutor: Luc Jaulin
Internship at Dobots 45 Stationplein 3013AK Rotterdam

September 20th 2016

Abstract

The prospects of home automation is having a whole new area where devices actively interact with the end-user and enhance his experience. For instance, for an inhabitant, lights can be turned off when he leaves the room or it can help a cleaning robot to find its way indoor. In order to executed these tasks, the devices need to locate us as well we need to locate ourselves. Using Bluetooth Low Energy signals, we are looking to have a room localization. The CrownStone is a device using Bluetooth Low Energy to communicate and we are looking to use these signals to localize ourselves indoor. But from a phone to another, the chip, the sensor and the hardware differ and induce major differences on the reads of the signals.

L'automatisation de notre intérieur nous permets d'avoir de nombreux appareils communiquant entre eux et fonctionnant de manière totalement autonome, augmentant notre confort et la facilité de vie. Mais de nombreux appareils n'ont pas cette capacité de se connecter sur le réseau de l'internet of Things. Crownstone est un adaptateur capable de se connecter à notre smartphone et nous donner la capacité de contrôler l'appareil sur lequel il est branché. Une de ses caractéristiques est d'utiliser des signaux Bluetooth Low Energy et nous souhaitons d'utiliser ces signaux pour pouvoir nous localiser dans le milieu intérieur. Mais d'un appareil à un autre, nous pouvons avoir des différences significatives dans la réception du signal.

Contents

Acknowledgment

I gives my thanks to the team behind DoBots and their collaboration for my work is hereby gratefully acknowledged.

I would also thank Ben,Stephen and Max for the awesome stay at their home.

About the company

Crownstone[1] was originally developed by Dobots which is a daughter company of Almende, a SME (Small and Medium-sized Enterprise) conceiving solutions in the ITC fields. It became it's own company in 2016 and specializes itself in home automation and its relation with the Internet Of Things. The core team stayed the same allowing a rapid and efficient transition.
With most of the team working in the same office and with a short hierarchic line, the company is able to operate and react quickly and keeps transparency. Regular feedback and planning meeting allow the team to work together while letting each employee to keep a large amount of autonomy in their work.

[1] https://crownstone.rocks/

List of Acronyms

BLE Bluetooth Low Energy
RSS(I) Received Signal Strength (Indicator)
FHSS Frequency-Hopping Spread Spectrum
dBm deciBel-milliWatt
SES Simple Exponential Smoothing
KF Kalman Filter

Chapter 1

Introduction

Over this last decade, the number of intelligent and connected devices skyrocketed. One of the most remarkable of them is the mobile phone which became a multimedia device with always growing computation power and connectivity.

This smart phone quickly became the most owned device. Its versatility and the numerous of built-in sensors make it the perfect hub to bind and connect the intelligent devices together.

The spread of intelligence and connectivity gave birth to new applications and in particular,in the field of home automation.

Home automation involves the control and automation of the different appliances. The Crownstone, a “smart” power outlet, allows to enhance the house comfort and control by connecting to traditional devices. The Crownstone works as a remote switcher, allows power monitoring and also works as a beacon for Indoor localization.

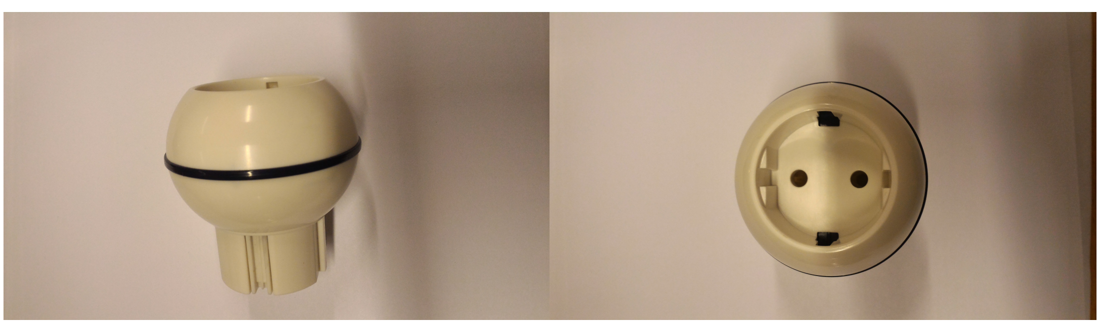

Figure 1.1: A Crownstone Prototype

1.1 Problem statement

Localizing ourselves indoor allows a great number of applications as turning on and off light depending of our presence, safe lock dangerous devices when out of reach. It could help robots and people to navigate indoors with few visual information.

Indoor localization with the Crownstone is based on Bluetooth Low Energy (**BLE**) signal strength. The Crownstone works as a beacon and communicates with the phone through Bluetooth. Knowing the position of the beacon, we can deduce the room we are in. But actually we quickly met technical limitations.

First, the BLE signal is prone to interfere with other electromagnetic signals and obstacles such the walls and people in the room and also is heavily influenced by the hardware and software on the phone.

To localize ourselves indoor, we could also use several methods such as odometry, using the signal strength to calculate the distance or measure the influence of the user's body (Device Free Localization). But fingerprinting was the method of localization kept as it is easier to implement and seems to be more viable and consistent than the other considered methods of localization. Fingerprinting is a method based on associating physical data, here the received signal strength indicator (**RSSI**) to a specific label, here an indoor area such as a room or a part of it. Using a collection of position-labeled fingerprints is a long process and needs a time-consumption calibration but at the end-point we gain in reliability and accuracy.

1.2 Tools

1.2.1 Phones

We are using the build-in Bluetooth sensors to read the Received Signal Strength. To read and store the data, I created a simple Android app that simply read the RSSI value from the BLE module and store it in a log file accompanied with a time stamp. The code source can be found in my GitHub[1]. To use it, the phone has to be compatible with Bluetooth Low Energy and the Android Version has to be 5 or higher. The application can be easily installed via Android Studio.

1.2.2 DoBeacons

DoBeacons are simple plug-in Bluetooth beacons. They contain only one sensor, a Bluetooth Low Energy sensor that can broadcast advertisements and be used to measure the RSS of nearby devices. In a realistic setting, one or two beacons

[1] github.com/MehdiN/bluetooth-fingerprinting

per room is good enough. We consider DoBeacons to broadcast uniformly in every direction.

Figure 1.2: A DoBeacon which is a simple BLE emitter

1.2.3 Gimbals

The sensor of the phone, the antenna, was not necessary isotropic. I made two two-axis gimbals to be able to measure the signal in every angle possible. I used LEGO Technic to build the main structure. More details about the gimbals will be given later in this documents.

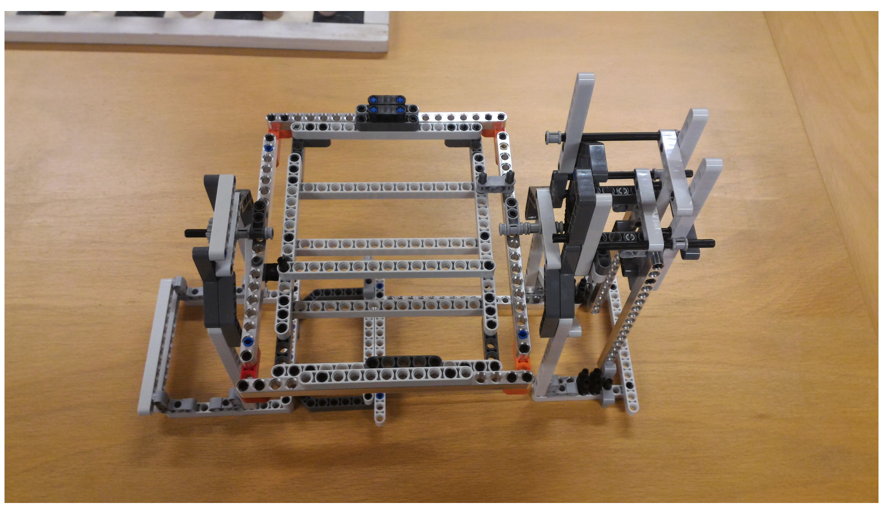

Figure 1.3: The first gimbal without its motors

Chapter 2

Bluetooth Specifications

2.1 Bluetooth overview

From the free encyclopedia, Bluetooth is a wireless technology standard occupying a section of the 2.4 GHz ISM band, shared with the WIFI protocol.

Bluetooth operated at frequencies around 2.4 GHz and uses **FHSS**, a radio technology allowing to divide data into packets and broadcasts each packet in a different channel. Bluetooth Low Energy accommodates 40 different channels

The Bluetooth technology used here is Bluetooth 4.X which enables BLE. We see with this version improvement with energy consumption and more options for a more secure broadcast.

2.2 Influential Factors on BLE signal

Bluetooth has a worse signal consistency compared to WIFI or other radio signals. The signals interfere with almost everything. But our end goal is to use it in an indoor environment so we have to include interference from other devices, WIFI and moving obstacles (e.g persons around the user).

2.2.1 Distance

As an electromagnetic radiation traveling through space, Bluetooth signal is being subject to attenuation by propagation. The path loss is usually expressed in dB and a value can be calculate with a log-distance model given with (2.1). $\overline{PL}$ is the average path-loss at the distance d, given with $PL(d_0)$ a known attenuation at the distance d_0 .

$$\bar{PL}_{[dB]}(d) = PL(d_0) + 10nlog(\frac{d}{d_0}) \tag{2.1}$$

Distance[m]	RSSI[dBm]	$\triangle RSSI[dBm]$
1	-52.4 ± 6.3	0
2	-57.8 ± 5.4	-5.4
3	-59.5 ± 5.1	-7.1
4	-63.9 ± 5.5	-11.5
5	-69.1 ± 5.7	-16.7

Table 2.1: Mean and Standard Deviation of RSS (Raw data without any filtering)

The inverse of this function can be used to determinate an approximation of the distance to the device.
The path-loss exponent is n=2 for free space. In practical case, the value of n might highly vary, obstructions and interference can lead to higher values of n. The table 2.1 shows the the mean and the standard deviation of the RSS from 1 to 5 meter. For n=2, a loss of 6 dB is to be expected. The RSS does decrease with the same magnitude as predicted and it has been confirmed in previous works [1].

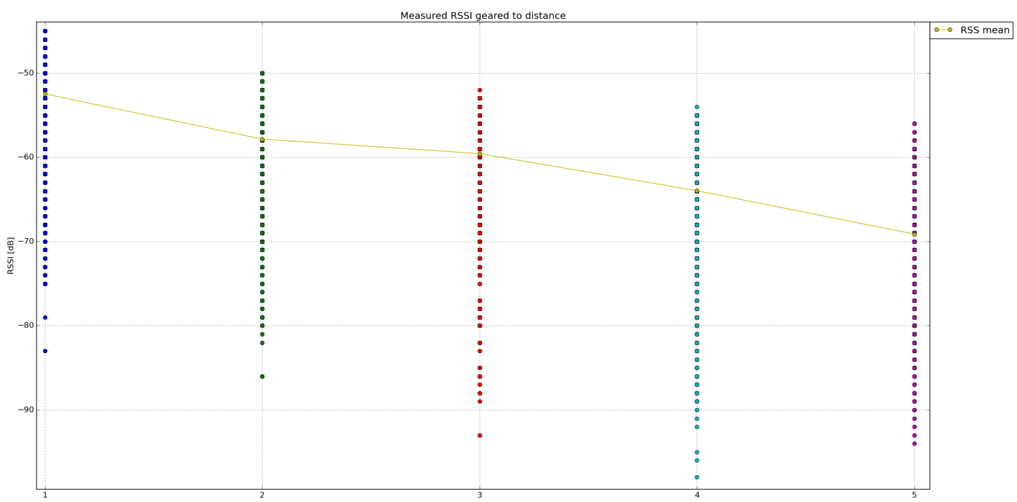

Figure 2.1: Measured RSS [No filtering]

2.2.2 Environment Influence

Depending of the material[2], the attenuation caused by this obstacle lies between 2 and 20 dBm. A simple concrete wall have an influence of more or less 2 dBm. For a human body, carrying the receiver or transmitter in the pocket, we can have an attenuation in the order of 5 to 20 dBm, depending on the distance to the devices.

2.2.3 Hardware Influence

The phone (receiver) has also an impact in the reception of the Bluetooth signal. The place where the antenna is embedded or the architecture of the chipset can influence the signal received.
We can see in the table 2.2 the change of signal strength between 2 phones used for our measurements.

Phone	RSS[dBm]	$\triangle RSSI[dBm]$
Asus Z0AD	-58 ± 1	$+4.9$
WIKO Lenny2	-56 ± 1	$+6.8$

Table 2.2: Change in RSS between two phones [Same Distance]

2.3 Filter Design

As we saw in the previous section, the Bluetooth signal is prone to interference from many sources. To read the signal better, I implemented two simple filters in order to obtain a more stable RSS through time. It is to be remarked that some manufacturers might implements their own filter in their device, same goes with some Bluetooth Scan Applications.

2.3.1 Normalization

I principally worked with devices with Android but we needed to compare the data from different phones and we can not exclude the smartphone of Apple. The IPhone BLE Bundle takes one sample per second while the majority of Android phones measures 3 to 10 samples per second.

We needed to normalize the data in other words to relate to one sample per second. I came by a very simple algorithm that takes the different values under 1 second frame and calculate the mean as the new value. Later, I came back with a second version, which I believe to be more precise and exclude aberrant sample. The algorithm is given in pseudo-code in the following.

Algorithm 2.1 Normalization

$[t_{min}, t_{max}]$ the 1 second frame
$L = [x_0, x_1..., x_n]$ the RSS values at t between t_{min} and t_{max}
for i in range(n):
$\alpha = mean(L)$
$\beta = std(L)$ #standard deviation
if $\alpha - \beta \leq x_i \leq \alpha + \beta$:
$x_i \rightarrow L_{new}$ #put the value in a new list

$x_{normalize} = mean(L_{new})$

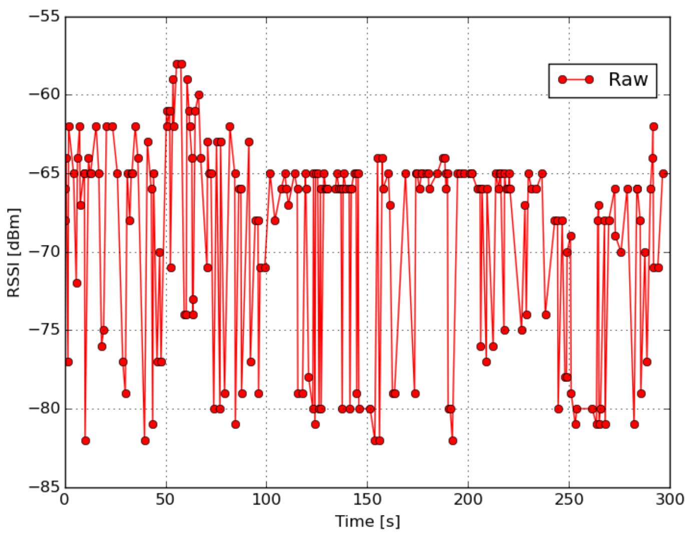

Figure 2.2: Raw RSS measured from a phone

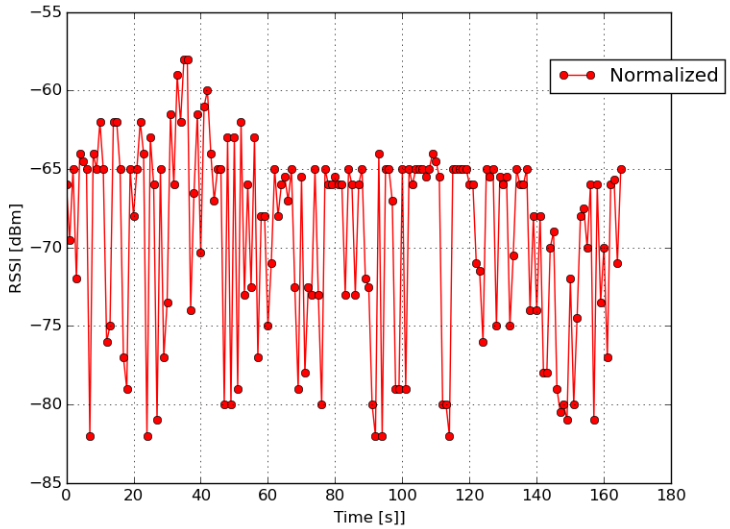

Figure 2.3: Normalized data from the same sample

2.3.2 Simple Exponential Smoothing

With the same aim of the normalization seen upper, the Simple Exponential Smoothing (SES) model allow to average and cut interference from the signal by giving more weight to recent data. In this fashion, the most recent observation has more weight than the 2nd most recent and the 2nd most recent has a bit more weight than the 3rd most recent and so on. The algorithm[3] is given in the following.

Algorithm 2.2 Brown's Simple Exponential Smoothing

Y is the measured signal and $\hat{Y}$ is the new value corresponding to the signal
L represents the current level (local mean value) and α denotes a smoothing constant between 0 and 1.
Step 1
$L_t = \alpha Y_t + (1 - \alpha)L_{t-1}$
Step 2
$\hat{Y_{t+1}} = L_t$
Step 3
$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t$

As we can see, setting the value of α allows us to set how much we want to filter.

	RSS[dBm]	$\triangle RSS[dBm]$	α
Raw data	-69.1	6.4	-
SES	-67.7	5.2	0.16
Normalized +SES	-67.9	1.8	0.16
SES	-68.8	6.6	0.66
Normalized +SES	-69.2	3.8	0.66

Table 2.3: Mean and Standard Deviation in RSS for different value of α

2.3.3 Kalman Filter

The previous filters helped to smooth the RSS lightly but we are looking for more stability in the signal. We need to suppress the remaining noise. The Kalman filter [3] is a state estimator allowing to make an estimation of the actual state based on noisy measurement.

As a recursive algorithm, it takes in account the past measurement and its reliability makes it that we can consider to include it in future Bluetooth Scan Applications for processing the signal while scanning. In the following we will establish the hypothesis needed to implement the algorithm.
The transition model has for general form the following:

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \tag{2.2}$$

$$z_t = C_t x_t + \delta_t \tag{2.3}$$

The current state x_t is defined as a combination of the previous state x_{t-1} and the control input u and the noise ϵ. Also the measurement is linked to the actual state by the second relation where δ is the measurement noise. A, B, C are matrix. In our model, we consider ourselves as static (we are not moving or not considerably to affect the signal i.e. staying in the same room). Also we ignore any control input in our system.

$$x_t \approx x_{t-1} + \epsilon_t \tag{2.4}$$

We also consider that our state is equal at our measurement:

$$z_t \approx x_t + \delta_t \tag{2.5}$$

Algorithm 2.3 Simplified Kalman Filter [5]

Prediction step

$\bar{\mu}_t = \mu_{t-1}$

$\bar{\Sigma}_t = \Sigma_{t-1} + R_t$

Kalman gain

$K_t = \bar{\Sigma}_t(\bar{\Sigma}_t + Q_t)^{-1}$

Update step

$\mu_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t)$

$\Sigma_t = \bar{\Sigma}_t - (K_t\bar{\Sigma}_t)$

μ describes the prediction and x the true value of the state. Σ defines the certainty of the prediction (variance of the prediction). R is the covariance of the system noise.

For the RSS measurement, we used a low value for R (R=0.008) as we suppose that most of the noise is caused by the measurement. Q is the variance of the measurement i.e the noise in the actual measurements. Q is set in function of the intensity of the measurement noise. In our case where we had number of interference around our devices, we took a value of Q around 1.5. For higher value of Q, the filter was not reactive enough.

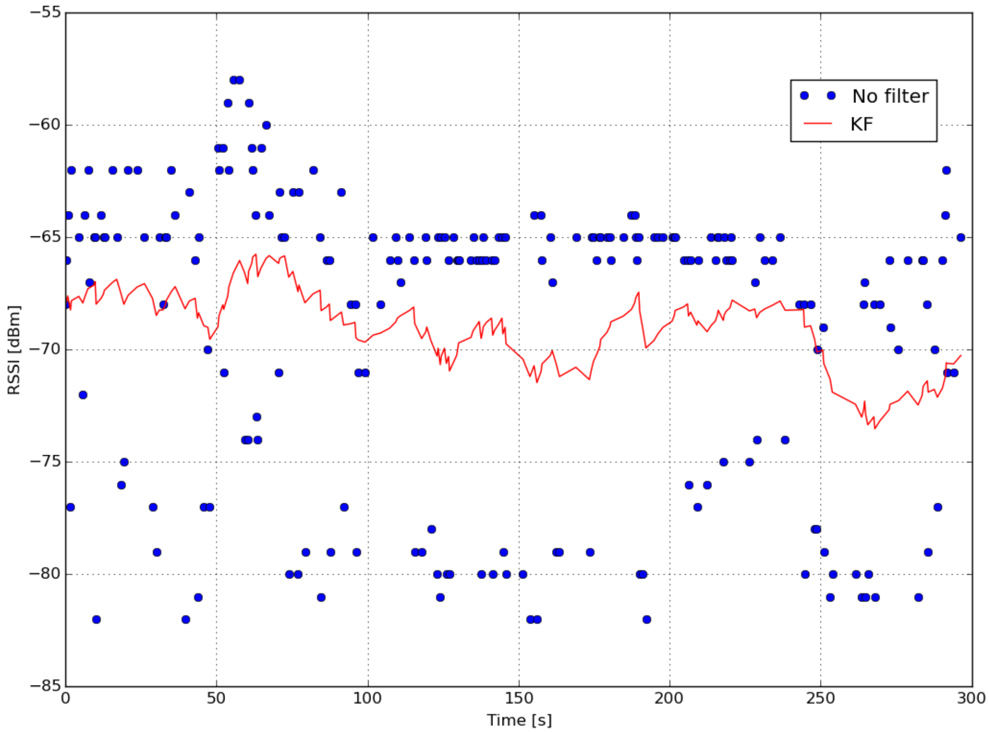

Figure 2.4: RSSI filtered with the Simplified Kalman Filter [R=0.008 and Q=1.5]

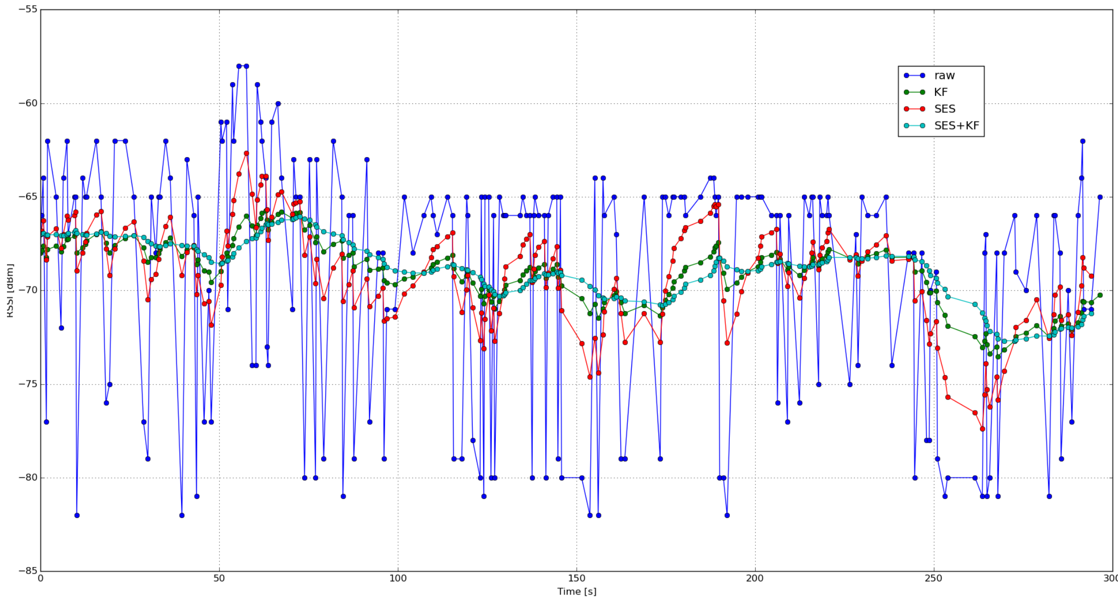

Figure 2.5: A look on the different filters

	RSS[dBm]	$\triangle RSS[dBm]$
No processing	-69.1	6.46
Kalman Filter only	-68.9	1.69
SES only	-69	2.66
KF + SES	-68.8	1.61

Table 2.4: Change in RSS mean and standard deviation for $\alpha = 0.195$ R=0.008 and Q=1.5

2.3.4 Data collection

I carried two series of measurements. The first one was for room level localization. The second one was to distinguish the difference between different phones and extract a general model which would be used to calculate the performance of the different device.
I used 2 to 4 different smartphones on Android to carry out our measurement. A self-made app was used to scan and get the RSSI from the different doBeacons. Collecting measurement took 5 min to 1 hour approximately, depending on the used method.

Chapter 3

Fingerprinting

The indoor localization we are trying to achieve does not necessitate the exact position of the user as we only need to know which room the user is in. Room level localization is likely to be the most suitable localization for our usage as we only need to trigger actions (e.g turning on or off the lights) when the user is or is not in a specific room. Having a room-level resolution in our localization makes our measurements easier as we only need to label the fingerprints, the measurements that will be used as a base for localizing the user, with the room where they were recorded.

3.1 Experimental Setup

3.1.1 Room localization

For this experiment, I divided the office into 3 rooms: Bedroom, Kitchen and Living Room. 2 doBeacons were attributed per room in exception of the Living room (the largest room) for which 3 doBeacons were attributed. The phones were put approximately in the middle of the room as we did not need to know the exact position of the user. There were no wall between the rooms. As we saw in section 2.2.2, the influence of a wall was around 2 dBm, it didn't matter in our experimentation.

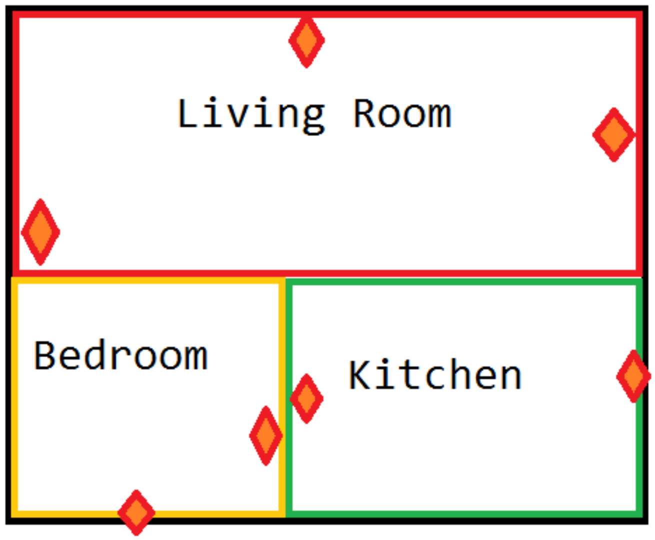

Figure 3.1: Floor Plan with the position of each DoBeacons

3.1.2 Fingerprinting on multiple phones

Each phone was inserted into a rotating platform. The platform, a two axis gimbal (or also called Cardan) whose each axis were motorized, allowed us to take RSS samples in almost every configuration of the phone. Usually, the user has the phone in his hand: the platform as large enough to attach other accessories to simulate others configuration. The idea to put a bloc of gel or a plastic glove with water to emulate the hand was suggested.
A two axis gimbal was enough for our need and allow to avoid the phenomenon of gimbal lock which happens for the 3 axis gimbals where they lose one degree of liberty.

The gimbals were made using LEGO Technic parts and they were controlled with a micro controller.

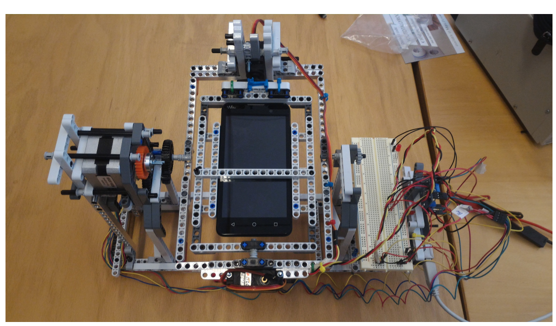

Figure 3.2: The first Gimbal with its motors

The first Gimbal was controlled with an Arduino MEGA. The outer platform was powered by a stepper motor and the inner platform was powered by a position servomotors. A motor shield was interfacing between the Step Motor and the Arduino board.

The axis of the outer gimbal was labeled x with a rotation angle θ. The inner gimbal was labeled with y and a rotation angle φ

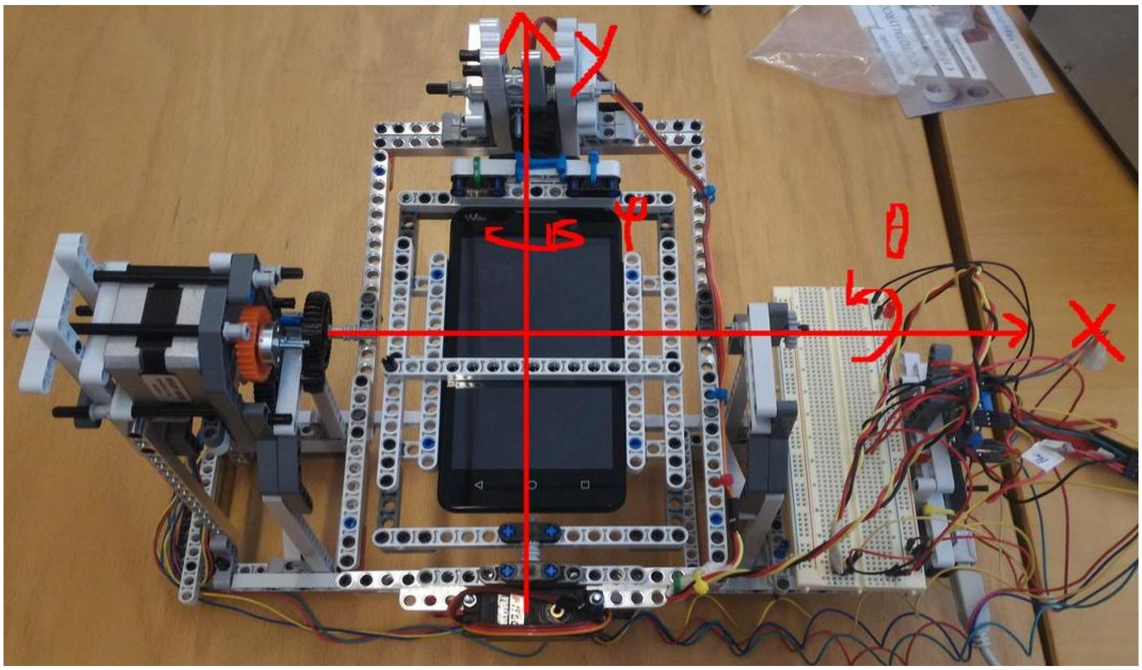

Figure 3.3: The different axis on the gimbals

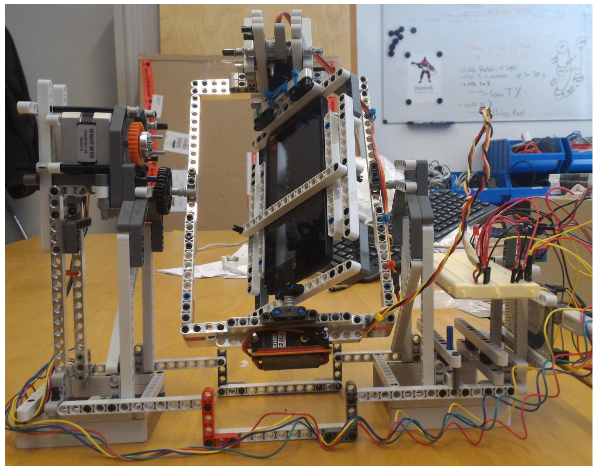

Figure 3.4: The phone is attached at the inner platform

The other gimbal was controlled by a NTX Lego controller and was motorized by two NTX Lego motors with Position feedback. The second one was less acurate in the positioning of the platforms.

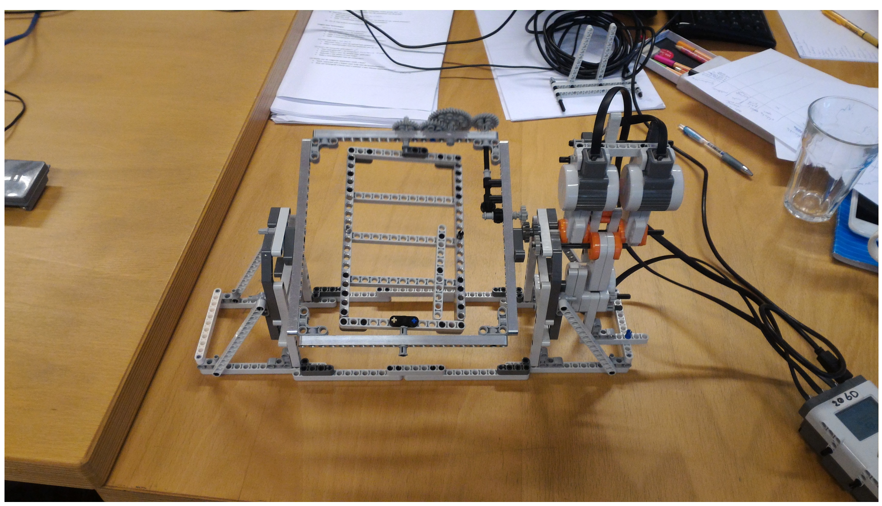

Figure 3.5: The Second Gimbal

3.2 Classification

3.2.1 Gaussian Naive Bayes Classifiers

To classify our fingerprints in order to predict and recognize further samples and localize where the user is, we made a simple hypothesis. The values associated to each class (one class per room) are supposed being distributed according to a Gaussian distribution.

The fingerprints x are divided and are attributed to a class c corresponding to a room of our choice. Then we compute the mean μ_c and the variance σ_c^2 associated to x. As we suppose we measure a new value v from an unknown class. Then the probability of v given a class c, $p(x = v|c)$ can be computed with

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}} \tag{3.1}$$

3.2.2 Phones Classifiers

If we succeed to establish a phone classifier, we can obtain a higher resolution in our localization. To do so, we tried to collect data and numerical values that could define the phone characteristics in regard of BLE signal.

Having these information could help in the scenario where the users do not have the same phone to localize themselves. Knowing which phone collected the fingerprints and which one is using using the data to localize itself would increase the fidelity in the prediction.

The measurement were made by attaching the phone on the gimbal. For each angular configuration, I collected samples for 90 seconds. I repeated the process for different distances. Every meter, a DoBeacon was placed in front of the gimbal, with a light shift to avoid to have all the beacons aligned. Not every DoBeacon was configured the same

Chapter 4

Results

4.1 Room Localization

The Asus phone is the one that collected the fingerprints.

Phone	Accuracy
Asus	>99%
Wiko	~66%
Lenovo	~90%

Table 4.1: Results with 7 beacons and no offset

As we can expect, the phone that collected the fingerprints has the higher accuracy. If we add an offset for the Wiko phone, we can obtain a higher accuracy. But when we are in the least favorable configuration (one beacon per room), our accuracy drops under 33% for almost every phone. Some beacons are at the limits between two rooms, as we can see in figure 3.1, which can confuse the prediction. I can deduce two important points:

First, for a low number of DoBeacons, the Gaussian Naive Bayes Classifier fails to be accurate enough to predict the right room.

Second point, the fingerprints may be not enough. With a more large database, we could obtain a higher accuracy. But we want the user to avoid collecting too much fingerprints as it is time consuming.
The answer can lie in different solutions. We can use other classifiers (e.g K-Nearest Neighbors). Dimensional reduction[6] is something which can be applied as well.

In parallel, establish a model of the receiver can increase the likelihood to find the right room.

Actually the room-level localization with the Crownstone is working with the Iphone.

4.2 Establishment of a phone model

Collecting fingerprints with several geometric combination of the phone position allow us to obtain a better view on the characteristic of the phone. Here we tested two phones from the same manufacturer and sold under the same name. The first one was from late 2015 to early 2016 and the second one was from mid-2016. As we can see in the two following figures, the signal received were different. The global conditions of the collecting process were the same and the same App and DoBeacons were used.

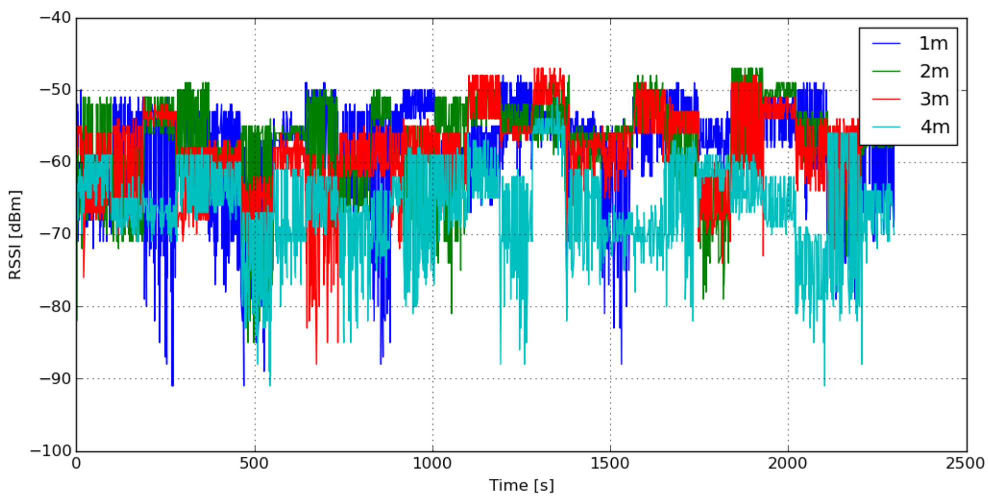

Figure 4.1: RSS of the phone n°1 [no post processing]

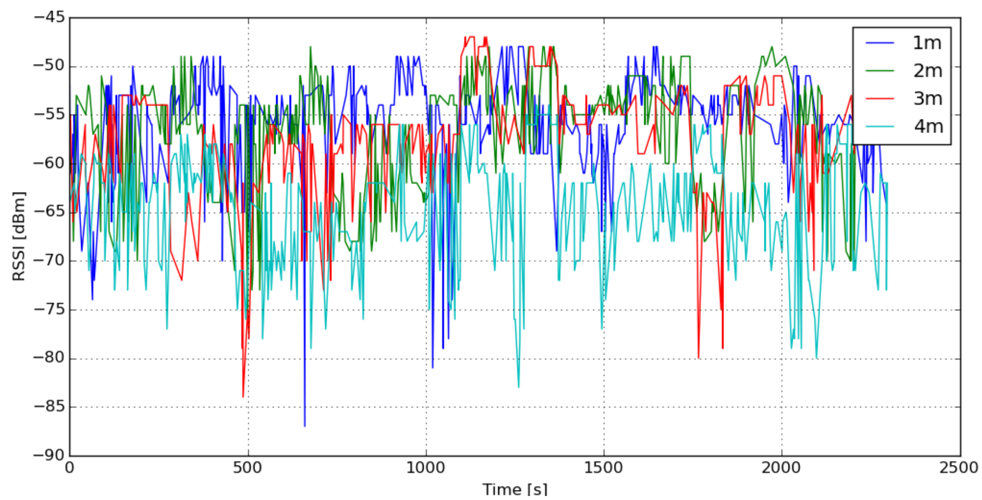

Figure 4.2: RSS to the phone n°2 [No post processing]

Then, we applied the two filters mentioned before (Simple Exponential Smoothing and the simplified Kalman Filter) making easier the read of the signal. We can notice one of the optimal configuration for receiving the signal for the phone 1 on figure 4.3. At 3m, the RSS is -49.5 dBm and the configuration of the phone is $\theta = 90°$ by the axis X and $\varphi = 135°$.

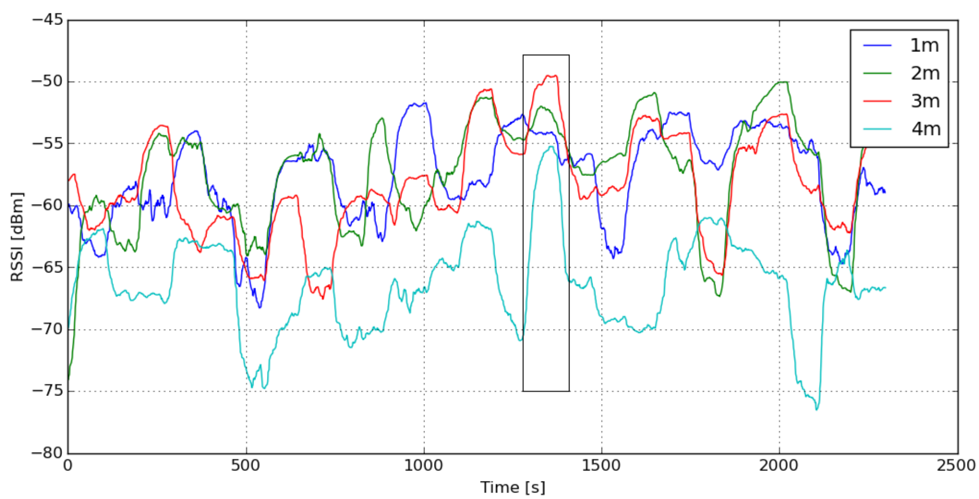

Figure 4.3: RSS of the phone n°1 [Filtered]

For the second phone, for 3m, we can find the same configuration, with a RSSI of 51.4 dBm.

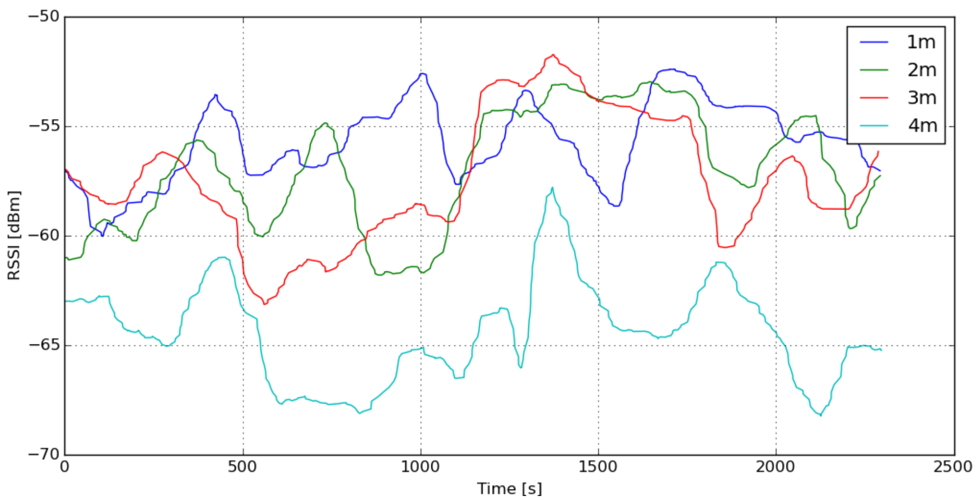

Figure 4.4: RSS of phone n°2 [Filtered]

I collected the different RSSI from 1m to 7m and look on the mean and standard deviation of the different dataset. The measurements could be improved if the same kind of beacons was used.

	Phone n°1		Phone n°2	
Distance	RSS [dBm]	$\triangle RSS[dBm]$	RSS [dBm]	$\triangle RSS[dBm]$
1	-57.5	3.6	-55.7	1.8
2	-57.3	4.2	-56.8	2.6
3	-58.0	4.1	-57.6	2.8
4	-66.2	3.8	-64.2	2.2
5	-61.2	4.0	-58.4	3.0
6	-69.4	3.0	-69.4	1.7
7	-68.5	2.6	-69.4	3.0

Table 4.2: Mean and Standard deviation translating the behavior of the 2 phones

Chapter 5

Conclusion

Room level localization can be achieved especially if we can outline the phone's characteristics.

The results showed that it is possible to find these parameters if the noise of the Bluetooth signal is deadened. The logical next steps would be:

-to find optimized parameters for filtering the signal.

-to find the parameters for phone classifier and model the behavior.

Playing with both room classifier and phone classifier would improve the fidelity of the fingerprinting method.

About the internship and self-analysis

To conclude this report, I will talk about the experience I gained from these 3 months abroad. This kind of internship was a first to me, and I lacked experience on many levels.

I would like to start with what I learned. What I mainly gained were not technical skills or neither language skill even if it is where I mainly improved during my stay in the Netherlands. The most valuable matter I learned there is where are my flaws.

- Mindset: even thought I prepared myself, I stayed a student. I have to go out of this comfort zone and think myself as a future engineer and accept to be an employee.

- Expressing myself. Good team work goes through communication. I did not express myself in a satisfactory fashion.

- Obstinacy. To not confuse with autonomy. Accept to go out my own way and follow the instructions.

On my professional project, an environment similar to the one offered by DoBots could be where I can thrive if I choose the business world : a short hierarchy chain, a balance between autonomous work and team work sessions seems to be most adapted for. But as I said, I need to grow more and work on myself.

Bibliography

[1] Tom van Dijk,"Indoor Localization Using BLE",p. 5, 2016

[2] J.C Stein "Indoor radio wlan performance par ii: Range performance in a dense office environment"

[3] Robert Nau Fuqua, "Statistical forecasting: notes on regression and time series analysis", School of Business Duke University

[4] Ramsey Faragher, "Lecture Notes", IEEE SIGNAL PROCESSING Magazine, p. 128, 2012

[5] Wouter Bulten, "Removing noise from RSSI signals", wounterbulten.nl ,2015

[6] Tom van Dijk,"Indoor Localization Using BLE",p. 38, 2016