CS-557

Cyber Physical Final Project

Indoor localization via Bluetooth Low Energy

Thibault Forest – Cyril Trosset – Thomas Theissier
11/29/2013

Contents

Background

Today, people can localize themselves almost wherever they want using GPS coordinates. To complement GPS, low signal in indoors places, some emerging technology (e.g.: iPhone 5) uses Wi-Fi localization. However, GPS doesn't perform well in indoor and dense urban areas even with the Wi-Fi localization. Actually, when people enter a building, there is no way to know their exact location. As a consequence, indoor localization is clearly an interesting subject in the sense that nobody has found solutions to know exactly where people are in a building at a precise instant. Indoor localization can be implemented in indoor areas using different sensors such as camera, FM signals, Wi-Fi, Bluetooth and others. Knowing that Bluetooth 4.0 with low energy technology paves the way for Bluetooth Smart devices, we decided to use this technology.

What about BLE?

Bluetooth 4.0 low energy was first introduced under the name Wibree by Nokia in 2006 and became part of the main Bluetooth standard in 2010, the year Bluetooth Core Specification Version 4.0 was adopted.

After this adoption in July 2010, many devices were under development using this technology. We have to wait for late 2011 to see the first Bluetooth Smart Ready mobile phone.

Features of Bluetooth Low Energy wireless technology include: ultra-low peak, average and idle mode power consumption, ability to run for years on standard coin-cell batteries, low cost, multi-vendor interoperability, and enhanced range.

These new devices have been appeared on several markets (health care, sports and fitness, security and home entertainment). A new tendency, which has come up with these BLE devices, also called beacon, advertises customer on their smartphones while they are shopping.

Project goals

Our project consists in localizing a Bluetooth device indoor, using iPhone and beacon technology.

1. Estimating the distance between a smartphone and a beacon device.
2. Compute the coordinates of the smartphone using three beacons.
3. Use additional sensors to improve the localization's accuracy.
4. Plot the path of a smartphone movement using Bluetooth devices.

Introduction

Indoor localization using iBeacons

Today, researchers are implementing some systems that allow people to localize themselves in indoor areas such as malls, shops, museums and many others. To implement these types of indoor localization, people uses technologies such as Wi-Fi, Bluetooth, cameras, FM signals and many others. We noticed that most of the indoor localization systems are not very accurate. We decided to focus on Bluetooth Low Energy after Apple released the new update of its operating systems (iOS7).

We chose this technology for several reasons. First and foremost, we thought that a brand new technology could allow us to be very precise on our localization. Second, a lot of companies were implementing the devices (the name of these devices is: beacons) while Apple was developing the new update of iOS in order to provide the beacons framework immediately after Apple released its new update of iOS. Third, because beacons are very affordable devices (those we bought was 30$ piece) and can be installed wherever people want to.

Few months ago were developed the first iBeacons. After Apple released its new update of iOS, we understood that iBeacons are devices that can emit at a range of 70meters (229 feet) and, as a consequence, detect iOS devices. At this point, we considered that with this power and using several iBeacons, we would be able to know the precise location of a person in a room of 70 square meters. And then by deploying multiple beacons in a building, we should be able to locate precisely a person. The applications are multiple: shops, museums, stadiums, and malls but also for individual like home automation.

To summarize everything and to provide a concrete example, our goal is rather than putting a beacons near each “trigger” (like paints in a museum), we want to provide a solution that provides the same functionalities with only 3 beacons per room and being able to act as an indoor localization solution in general which is not possible with the basic proximity functionality.

Our choice: Estimote

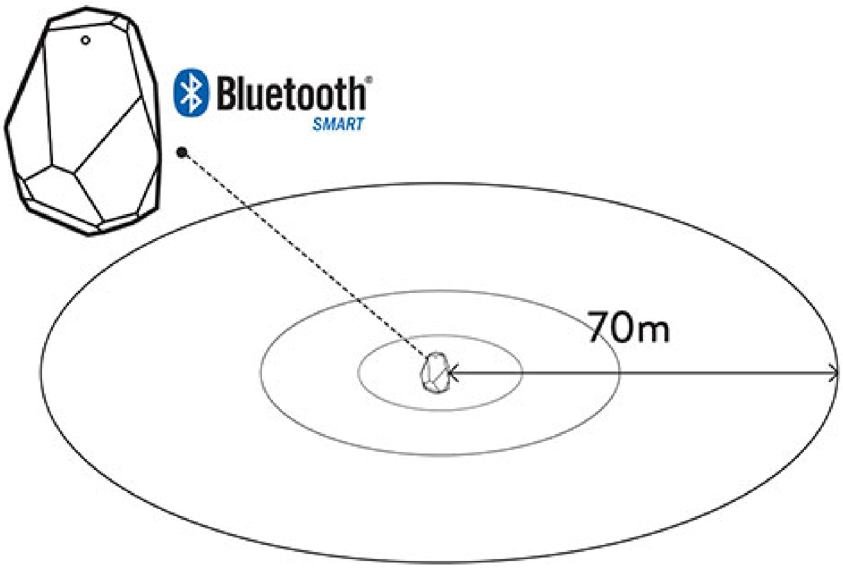
Choosing the hardware for our project was not as easy as it seems to be. We decided to search for a dedicated solution. Our options were:
- a BLE dongle running on a virtual machine. But it was not convenient to use one computer per beacon needed. And it will have limited our testing possibilities.
- a Raspberry PI or Arduino-based solution. Nevertheless, the cost per beacon would have been too high. Moreover it would have implied hardware design overhead.
- a dedicated solution.

However, since the technology was new, the potential beacons suppliers were extremely restrained. Fortunately, we were able to find some: (Roximity, Estimote, RedBearLab, Tile).

We chose Estimote for several reasons.
First, Estimote was the only company, which was able to ship the beacons according to our timeframe and, at an affordable price (99$ for three beacons). Second, we paid particular attention to what was provided with these beacons. Since we wanted to implement an iOS application, an iOS API was required to allow the communication between the iOS program and the beacons. Fortunately, Estimote provided an API with tutorials and full documentation, and is compatible with the iOS 7 beacons framework, which help us to understand how it works. Third, we were interested in buying 3 beacons to implement our system and the Developer’s Preview kits of Estimote included a trio of beacons, which was the perfect opportunity for us to buy beacons.

How iBeacons work

A beacon can be seen as a super small computer. In fact, it is composed of a 2-bit ARM® Cortex M0 CPU with 256kB flash memory, an accelerometer and a temperature sensor, and 2.4 GHz Bluetooth 4.0 Smart (also known as BLE or Bluetooth low energy) bidirectional radio. Depending on the signal strength, a beacon can run up to 2+ years with a single coin battery (CR2450, 20mA/h).



Phones or other smart devices in the range can pickup the Bluetooth radio signal (without previous pairing) and estimate their distance to the beacon by measuring the received signal strength (RSSI). The closer you are to the beacon, the stronger the signal is.

Estimote beacons can be uniquely identified as well since each beacon broadcasts its own id. Every ID is 20 bytes long and is divided into 3 sections:

$$\boxed{ProximityUUID\ (16\ bytes)\ +\ major\ number\ (2\ bytes)\ +\ minor\ number\ (2\ bytes)}$$

ProximityUUID is a unique UUID to distinguish beacons from those others are using. The major number is used to group related set of beacons. Minor number is used to identify a beacon within a group of beacons. In our case, UUID and major are the same for our 3 beacons.

To better understand and summarize how you can use these, you might want to use one UUID for an entire company and then have Major be the individual stores. Minor could specify individual shelves inside each store.

Localizations methods

Distance estimation

As explain above, we use an iPhone to detect a beacon. In order to obtain the distance between these two devices, we measure the RSSI signal emitted by the beacon, and received by the iPhone.

RSSI stand for received signal strength indication. It is a measurement of the power of a telecommunication signal, Bluetooth signal in our case.

An iPhone is able to measure an RSSI Bluetooth signal using the appropriate framework (CoreBluetooth & CoreLocation). The value measured is returned in dBm.

Below the formula to convert a distance into a RSSI signal:

$$RSSI_{dBm} = -(10 \times n \times \log(d) - A)$$

Where n is a constant, d is the distance in meter, and A the RSSI value for one meter.

However, we needed the reverse equation, that is to say converting a RSSI signal into a distance:

$$d = 10^{-\frac{RSSI_{dBm} - A}{10 \times n}}, \quad A \approx -62dBm \approx 1\, meter, \qquad n = 43.19$$

Mathematical methods

There are several methods to calculate points coordinate using distances or angles. Triangulation, trilateration, multilateration are well-known methods. Whereas triangulation uses angles to compute the position of a point, trilateration works with distances. The latter is obviously the one which is more appropriate for our project.

Trilateration

Trilateration is used to determine absolute or relative locations of points by measurement of distances, using the geometry of circles, spheres or triangles. This method has practical applications in surveying and navigation (GPS for example).

Below an example of trilateration computation:

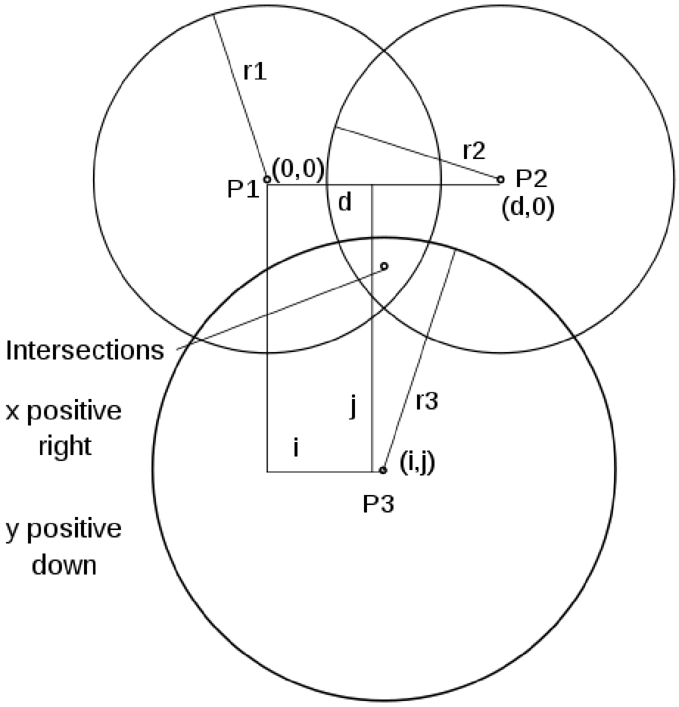


Figure 1 Trilateration scheme

P1, P2 and P3 can be assimilated as beacon devices. Their coordinates are known. r1, r2 and r3 are respectively the radius of P1, P2 and P3 spheres.

How to find the coordinates of the intersection (x, y, z)?

Below formulas of the three spheres:

$$r_1^2 = x^2 + y^2 + z^2 \ (1)$$

$$r_2^2 = (x-d)^2 + y^2 + z^2 \ (2)$$

$$r_3^2 = (x-i)^2 + (y-j)^2 + z^2 \ (3)$$

Solve x, (1) - (2):

$$x = \frac{r_1^2 - r_2^2 + d^2}{2 \times d}$$

Solve y, substituting the x equation back into (1):

$$r_1^2 = \frac{(r_1^2 - r_2^2 + d^2)^2}{4 \times d^2} + y^2 + z^2 \quad or \quad y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{4 \times d^2},$$
$$where\ z^2 = r_1^2 - x^2 - y^2$$

$$Then, y = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2 \times j} - \frac{i}{j} \times x$$

$$Finally, z = \pm\sqrt{r_1^2 - x^2 - y^2}$$

To sum up:

$$x = \frac{r_1^2 - r_2^2 + d^2}{2 \times d}, \qquad y = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2 \times j} - \frac{i}{j} \times x, \qquad z = \pm\sqrt{r_1^2 - x^2 - y^2}$$

Thus, we have the (x,y,z) coordinates of the smartphone. We will not focus on the z-coordinate according to our project scope however it could be an extension of our project and could reduce the number of beacons required and/or increase the accuracy.
It turns out that the precision is not that accurate, moreover, what about if one beacon becomes unavailable? Can't we estimate the position with other low-consumption sensors?

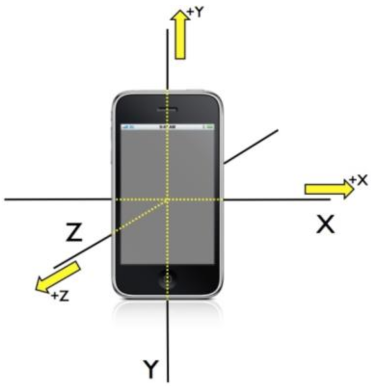
Step counter

A new feature designed by Apple in the iPhone 5S enable developers to easily get the number of step measured by the accelerometer and using a new processor called M7. Its particularity is to consume very low amount of energy and enable several applications to monitor user motion without degrading the battery life. Unfortunately, this feature does not allow us to get an instantaneous monitoring of the steps, but only to obtain an historic of the steps during a time interval. Actually we were able to detect the number of steps only each 7 steps. That is why we had to implement our own step counter using the iPhone accelerometer.

Accelerometer

An accelerometer is a sensor designed to measure the g-force acceleration (rate of change of velocity). These kinds of sensors are used to detect and monitor vibration; they are implemented in a lot of devices such as tablet computers, digital cameras, video game controllers, smartphones, and others.

The iPhone's accelerometer gives us the three dimensions x, y and z acceleration measured by the sensor.



In order to measure the total acceleration whatever the orientation of the smartphone, we use the following formula:

$$Y_t = \sqrt{x^2 + y^2 + z^2}$$

In addition to that we had to know the average acceleration produced by the user motion. A famous method called exponentially weighted moving average (EWMA) is perfect to get a dynamic average, meaning that the average will adapt its value quickly whatever the conditions in which the user is (slope).

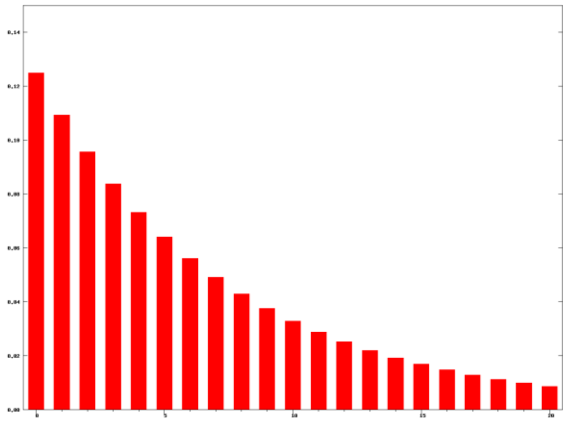


Figure 2 EWMA weight bar chart coefficient

$$EWMA\ calculated\ recursively:\ S_t = \propto \times Y_t + (1 - \propto) \times S_{t-1}\ ,\quad for\ t > 1$$

Where α represents the degree of weighting decrease. A higher α discounts older measures faster ($0 \leq \alpha \leq 1$). Y_t is the acceleration value at time t. S_t is the EWMA at time t. We decide to initialize $S_1 = Y_1$.

As drew in the graph below, the green line represents (Y_t), the instantaneous acceleration value. The blue line is the average of the green line (S_t). And the red line represents the number of steps.

Figure 3 Explicative graph of step counting

How we used these values to detect a step?

One step is defined with three stages:

1. The green line reaches the minimum peak.
2. Then crosses the blue line (EWMA).
3. Finally reaches the maximum peak.

A step is detected if the acceleration value follows these three stages. We set up a threshold (minimum and maximum), constant value which enables us to configure the sensitivity of the step counter. In our case the user hold his smartphone in his hand. The step counter has to be high sensitive. In our experiments we consider each step as 0.5 meter.

Compass

Magnetometers are widely used for measuring the Earth's magnetic field, in geophysical surveys to detect magnetic anomalies of various types; they can also be used as metal detectors. Nowadays a lot of devices have built in magnetometers.

In the iPhone, the magnetometer is a magneto-resistive permalloy sensor, it uses three-axis, meaning that it is not sensitive its orientation or elevation. One advantage is that it works outdoor and indoor. We use this integrated compass to get the user direction, considering that the user hold his smartphone following the direction he moves. We haven't considered that the user will hold his smartphone horizontally to simplify the implementation.

Unfortunately, we notice some differences between an original compass and the iPhone's compass. We approximate this error in a range of 0 to 20 degrees (margin of error varies when the iPhone recalibrate its compass).

Methods combination

In order to obtain the best accuracy and to encounter the case for which the user detects only two beacons, we combine the four technics described above.
First we use trilateration to determine the position of the user. By doing an average we reduce the interferences and adjust the position until a step is detected. When a step is detected, given the estimation of a step length and the measurement of the compass, we are able to predict a new position by computing Δx and Δy.

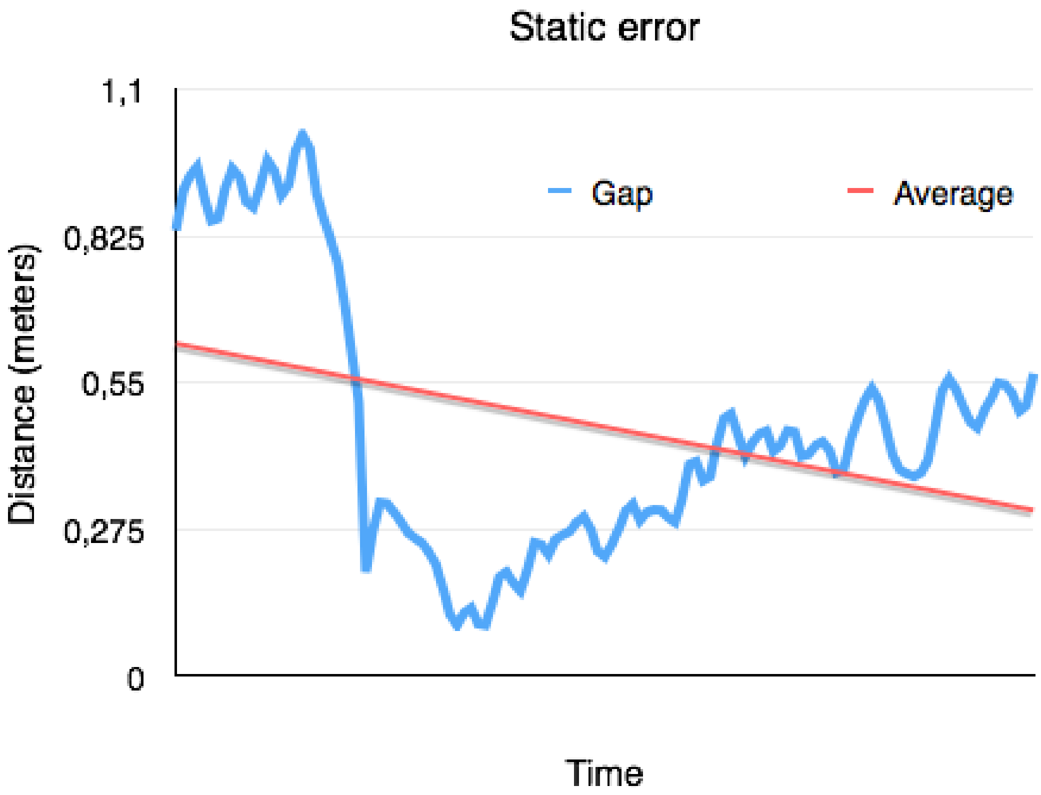
$$\Delta x = D * \sin(H * \frac{\pi}{180}) \qquad \Delta y = D * \cos(H * \frac{\pi}{180})$$

With: D: step length (meter) H: Gap from north (degree)

So we add Δx and Δy to the previous position, and apply a strong weight (0.6) on this value as it take benefit from the previous averaged trilateration. From this point we continue to compute the average of trilateration-position to adjust the "calculated" position until a new step is detected.

In doing so we are able to estimate the position whether or not the user is in the range (if previous positions are available) and take advantages of all previous computations. Meaning that interferences are lowered not only by average but also by adding some “trusted” sensors with acceptable accuracy and less subject to interferences.

Accuracy



The graph above represents the values measured via our system when the smartphone stays at a fixed point. The blue line pictures the gap between the real position and the computed position. The red line performs the gap average. This chart demonstrates that the position accuracy increases with the time. These results prove that performing an average on the measured values improve the exactness filtering interferences.

We did several simulations of a moving user in a room. The curves above illustrate the difference between real measures and our system's measures for a path. Before the yellow line starts the user is static. The graph shows an increase in the accuracy during the movement, because the blue and the yellow line come close to each other. The precision improvement while moving emphasizes the effectiveness of our system, which uses additional sensors as the smartphone is moving.

Conclusion

We have developed an innovative method using a brand new technology. Our technology is able to locate a user in a room and then can trigger some events. The latter part is the main purpose of the beacon concept, however this requires one beacon per trigger. We can confidently say that our solution presents good results and can be utilized in various environments. Nevertheless, it turns out that the interferences are more subsisting than expected, especially with our hardware (compare to an iPhone turned into a beacon). As a consequence, the localization is not very precise and can't equal the one-beacon-per-trigger performances. As an estimote beacon is barely larger than a coin and lasts two years, it is highly probable that augmenting the dimension and the emission power can result in better performances keeping a long battery life. At least we could try other beacons manufacturers as the result with estimote are promising but not satisfying for all conditions (for example crowded mall). Nonetheless, the beacons availability on the market will expand in several months and it will be an opportunity for us (selecting the more appropriate hardware) that we couldn't afford according to our project timeframe.