BLE Indoor Localization with Estimote Beacons

Umberto Fazio, Luca Fucci

May 15, 2018

1 Introduction

In this project we tried different techniques for localizing a Raspberry Pi in an indoor environment using BLE signals provided by Estimote Beacons. We applied trilateration and fingerprinting, which we'll discuss below, both in a static and dynamic framework, performing tracking through Kalman Filter. We tested the algorithm either in 2D and 3D, the results are reported below.

2 Data Collection

The algorithm has been tested with 8 Beacons in a $20m^2$ room. We collected RSSIs coming from the Raspberry Pi in 51 different position in the (x, y) plane for 3 different z values. At each position we gathered 150 measures, for a total of 183600, each of them annotated with the actual position. A NodeRED server scans the environment looking for Beacons, then it sends the measurement to an UDP stream, which will be read by our algorithm

3 Algorithm

3.1 Trilateration

First, we fitted a logarithmic model which relates the distance to the RSSI in our environment. Then, for each incoming RSSI, we estimate the distance from each Beacon using the formula

$$S = S_0 - 10\alpha \log_{10} \frac{d}{d_0} + v$$

Finally, we combine a batch of different measures in a linear system

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = d_i^2$$

which can be espressed as

$$2 \begin{bmatrix} x_n - x1 & y_n - y1 & z_n - z1 \\ \vdots & \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} & z_n - z_{n-1} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (d_1^2 - d_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) - (z_1^2 - z_n^2) \\ \vdots \\ (d_1^2 - d_n^2) - (x_{n-1}^2 - x_n^2) - (y_1^2 - y_n^2) - (z_{n-1}^2 - z_n^2) \end{bmatrix}$$

$$Ax = y$$

from which we estimate (x, y, z) by computing the Least Squares solution:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = (A^T A)^{-1} A^T y$$

3.2 Fingerprinting

Then we apply a fingerprinting estimation: we use the previously collected data with the position and RSSI for each Beacon to determine the next positions. The dataset is structured this way:

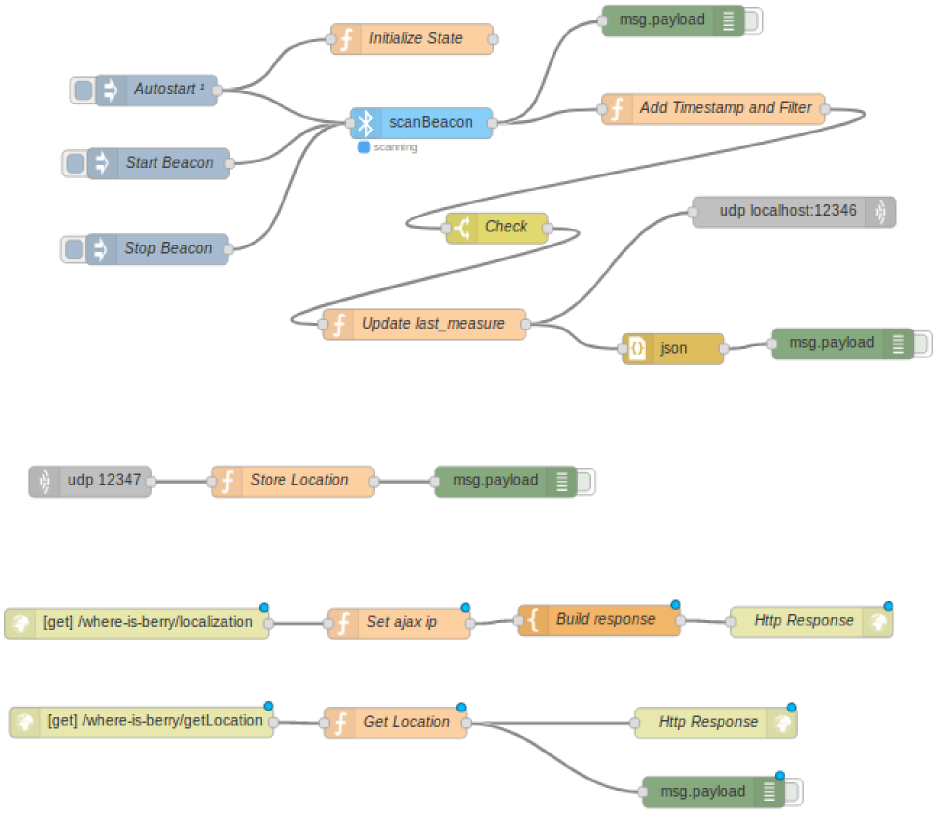

Figure 1: NodeRED schema. It includes the Beacon scanning through the Bleacon library and a simple API to visualize the data in a browser.

x, y, z	$\begin{bmatrix} RSSI(x,y,z)_1 & RSSI(x,y,z)_2 & \dots & RSSI(x,y,z)_8 \end{bmatrix}$

The algorithm waits to receive at least one RSSI for each beacon. Then, it performs a K-Nearest Neighborhood to determine which are the positions for which the RSSI are most similar, and then we compute an average of their positions, weighted by their RSSI.

3.3 Tracking

The RSSI signal is filtered via Kalman Filter, for which we provide the formulation in our setting:
Prediction step

$$\hat{x}(t) = Fx(t-1)$$
$$\hat{P}(t) = Fx(t-1)F^T + Q$$

Measurement step

$$\hat{y}(t) = z(t) - H\hat{x}(t)$$
$$S(t) = H\hat{P}(t)H^T + R$$

Update step

$$K(t) = \hat{P}(t)H^T S(t)^{-1}$$
$$x(t) = \hat{x}(t) + K(t)\hat{y}(t)$$
$$P(t) = I - K(t)H\hat{P}(t)$$

State variables:

$$x = \begin{bmatrix} x_1(t) \\ \dot{x_1}(t) \\ x_2(t) \\ \dot{x_2}(t) \\ \vdots \\ x_8(t) \\ \dot{x_8}(t) \end{bmatrix}$$

where $x_i(t)$ is the RSSI of Beacon i at time t
Dynamic Model:

$$F = \begin{bmatrix} 1 & \Delta t & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \Delta t \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Process noise covariance matrix:

$$Q = \phi \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & \dots & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & \dots & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}$$

Measurement noise covariance matrix (computed from the data):

$$R = \begin{bmatrix} 60 & 0 & \dots & 0 \\ 0 & 60 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 60 \end{bmatrix}$$

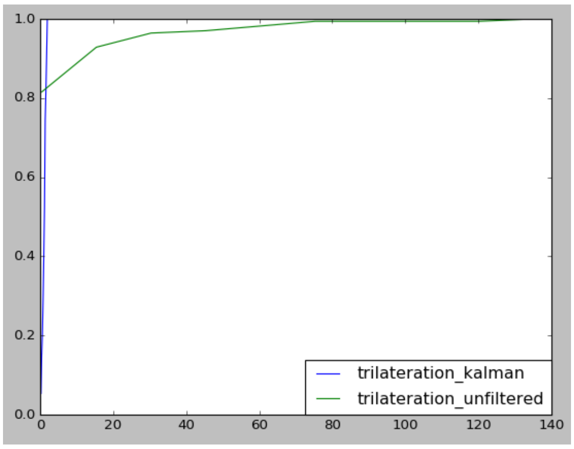

Figure 2: Trilateration CDF

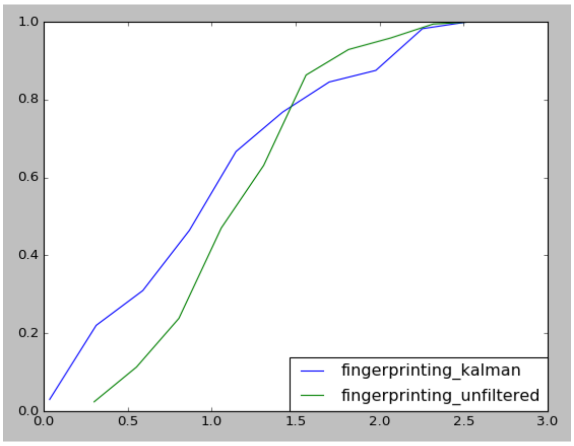

Figure 3: Fingerprinting CDF

4 Results

We tested the performance of the algorithm choosing 15 position from the dataset. Such positions have been choosen in a way so that they emulate a realistic path in the environment. In the figure 2 and 3 are the Cumulative Distribution Functions of the Trilateration and Fingerprinting techniques, either filtered or unfiltered. The error is computed as the Euclidean distance between real position and estimate positions. Here are the mean errors:

Technique	**Mean error**
Trilateration	11.97 m
Trilateration Filtered	1.28 m
Fingerprinting	1.38 m
Fingerprinting Filtered	1.26 m

References

[1] Subhan, Hasbullah, Ashraf *Kalman Filter-Based Hybrid Indoor Position Estimation Technique in Bluetooth Networks* https://www.hindawi.com/journals/ijno/2013/570964/

[2] Zafari, Papapanagiotou, Devetsikiotis, Hacker *An iBeacon based Proximity and Indoor Localization System Faheem Zafari, Ioannis Papapanagiotou, Michael Devetsikiotis, Thomas Hacker.*