Indoor Positioning System Using Bluetooth Beacon Technology

Bachelor Thesis in Computer Science

Mälardalen University
Academy of Innovation, Design and Engineering
Author: David Torstensson
Date: 2016-09-25
Examiner: Mats Björkman
Supervisor: Hossein Fotouhi
ABSTRACT

Bluetooth Low Energy (BLE) Beacons are transmitter devices that broadcast signals, which can be picked up by nearby devices using Bluetooth. Most commonly they are used for advertising, and for informational purposes. These devices can take advantage of Received Signal Strength Indication (RSSI) information together with an appropriate localization algorithm, to pinpoint a user’s position. This is particularly useful in indoor environments where GPS is no longer an option. Compared to other indoor positioning systems (IPS), BLE Beacons is a low-cost, low-power option.

This work provides a state-of-the-art on the positioning algorithms used by BLE Beacons, and identifies common error factors. Two common indoor positioning systems, GoIndoor and Navigine, are implemented and evaluated.
## Table of Contents

Abstract ................................................................................................................................. 1

1. Introduction ......................................................................................................................... 3
   1.1 Problem formulation ......................................................................................................... 3

2. Research Context ............................................................................................................... 4
   2.1 BLE Beacon Technology ............................................................................................... 4
   2.2 Received Signal Strength Indication ............................................................................... 4
   2.3 Protocols ....................................................................................................................... 5
      2.3.1 iBeacon .................................................................................................................. 5
      2.3.2 Eddystone .............................................................................................................. 5

3. Methodology ..................................................................................................................... 6

4. Literature review ............................................................................................................... 7
   4.1 Factors affecting positioning accuracy .......................................................................... 7
   4.2 Node Placement ............................................................................................................ 8
   4.3 Methods for distance estimation and noise reduction ................................................... 9
      4.3.1 Propagation model ................................................................................................. 10
      4.3.2 Kalman Filter ........................................................................................................ 10
      4.3.3 Window size .......................................................................................................... 11
   4.4 Localization techniques ............................................................................................... 12
      4.4.1 Fingerprinting using Euclidean Distance Correction Algorithm ............................ 13
      4.4.2 One dimensional positioning using three alternative approaches ......................... 14

5. Evaluation ......................................................................................................................... 17
   5.1 Setup ............................................................................................................................. 17
      5.1.1 Hardware ................................................................................................................ 17
      5.1.2 Software ................................................................................................................ 17
      5.1.3 Test Areas ............................................................................................................. 18
   5.2 Small Area .................................................................................................................... 19
   5.3 Kitchen environment .................................................................................................... 20
      5.3.1 Centralized beacon approach .............................................................................. 21
      5.3.2 Wall mounted beacon approach .......................................................................... 22
      5.3.3 Microwave test ...................................................................................................... 23
   5.4 Hallway ........................................................................................................................ 24
      5.4.2 Fingerprinting ....................................................................................................... 25

6. Conclusion ......................................................................................................................... 26
   6.1 Suggested improvements .............................................................................................. 26

References ............................................................................................................................ 28
1. INTRODUCTION

One of the most common ways of locating a person or a mobile device, is using GPS hardware because of the availability of this technology in smartphones. In an indoor environment without satellite connection, GPS technology is no longer an option. In such scenarios an indoor positioning system (IPS) is often employed. For indoor positioning there are many different approaches and technologies, but there is a lack of standard algorithms and hardware.

BLE Beacon Technology has developed during the last few years, as a means of sending position-based data to nearby users. Apple’s iBeacon and Google’s Eddystone are both examples of these small devices that can be attached to almost any surface, and then broadcast a signal which can be picked up by smartphones and other devices that utilize Bluetooth. Typically, BLE devices are used for advertising and informing users of nearby points of interest. However, using RSSI information, an application can estimate the distance to a beacon. Consequently, with the use of three or more beacons together with an effective algorithm, the user’s position can be pinpointed.

This work will evaluate the performance of some common algorithms for positioning that utilize BLE Beacon Technology. Beacon Technology has some benefits in being low-cost and low-power. Beacons were primarily built to detect proximity, but are also used for localization. However, there are multiple factors that negatively affect the accuracy. Thereby, Beacons often find their use case in indoor positioning as a complement to other technologies [1]. Bluetooth Beacons can for example be used as a complement to WiFi, which gives better accuracy than using only WiFi [4]. The accuracy of Beacons can however be significantly improved using effective algorithms [2, 3].

1.1 Problem formulation

The algorithms for determining a user’s position in an indoor positioning system (IPS) significantly affects the accuracy of the results. An effective algorithm moderates and in a best case scenario cancels out factors that negatively affect the positioning result. The purpose of this thesis is to research which algorithms are currently used for indoor positioning using BLE Beacons, and to evaluate one of these algorithms in a real-world experiment with regard to different error factors. The following research questions will be answered:

- Which are the common positioning algorithms for BLE Beacons?
- Which factors can affect the positioning result?
- How effectively can error factors be cancelled out using a common algorithm?
2. Research Context

This section is an overview of the technical aspects of Beacon Technology. 2.1 describes the hardware aspect, 2.2 describes received signal strength indication (RSSI) which is used for positioning, and 2.3 is a technical specification of the two available beacon protocols, iBeacon and Eddystone.

2.1 BLE Beacon Technology

Bluetooth Low Energy (BLE) Beacons, are small devices that can be applied to almost any surface, and communicate messages to nearby devices using Bluetooth. The technology mainly finds its use within advertising, and informing smartphone users of nearby points of interests.

A BLE Beacon broadcasts small packets of data, with a certain interval. The maximum payload of a Bluetooth 4.2 packet is 257 bytes, which is not enough to embed media content. Instead a beacon simply broadcasts a unique ID, and the application on the receiving device must recognize the beacon and perform relevant tasks. This is one-way communication, since beacons just broadcast signals and does not receive information.

Bluetooth Low Energy operates in the 2.4 GHz license-free band (ISM band), which is the same frequency range as used in WiFi Transceivers. This can cause some interference, resulting in packet loss, which can affect localization accuracy. On the other hand, a solution which utilizes both WiFi and Bluetooth simultaneously, gives a better localization accuracy than using either technology individually.

Beacon Technology is a relatively low-cost solution, a single beacon generally ranging from 100-400 SEK, depending on how advanced the model. It is also a low-power device, as a beacon can have a lifespan of several years [5]. Availability is an advantage, since its features can be utilized by anyone with a smartphone.

Broadcasting power can be adjusted on most beacons. This is a way to calibrate beacons for optimal distance estimation and positioning. For example, an Estimote Beacon has a power ranging between 4 dBm and -40 dBm, and the kontakt.io beacon ranges between 4 dBm and -30 dBm.

2.2 Received Signal Strength Indication

Bluetooth lacks precise time synchronization[19], which rules out time based triangulation methods. Instead, most Bluetooth based positioning systems rely solely on Received Signal Strength Indication (RSSI) for positioning.

RSSI is an indication of the strength of a Bluetooth signal when picked up by a receiver. This indicator is an 8-bit value ranging from -127 dBm to 20 dBm, which represents the power level in mW (milliwatts). The values are most commonly between -30 dBm and -102 dBm, where lower negative values identify stronger signals. -30 is considered a very good signal, which requires the receiver to be very close to the source, -70 is a low quality signal considered the minimum for reliable packet delivery, and -90 is a very poor signal, that is not recommended for data communication.

The relation between distance and loss in signal strength makes it possible to make a distance estimation. The RSSI can then be used to estimate a user’s position, using three
or more beacons, and IPS software which implements an RSSI based localization algorithm. The signal strength can however vary greatly, and requires effective filtering to stabilize the estimation results.

2.3 Protocols
There are two protocols for BLE Beacons available, Apple’s iBeacon and Google’s Eddystone. iBeacon has been widely used in the market, and most of Beacon devices support this protocol, whereas Eddystone is supported by a minority. Recently developed Beacons commonly support both of these two protocols. It is important to note that iBeacon and Eddystone are only protocols, whereas the hardware is developed by third party companies with a variety of shapes and technical specifications.

2.3.1 iBeacon
iBeacon was introduced by Apple in 2013. An iBeacon packet is up to 31 bytes, and consists of the following sets of data:

- **iBeacon Prefix** (9 bytes)
- **UUID** (16 bytes)
- **major number** (2 bytes)
- **minor number** (2 bytes)
- **TX Power** (1 byte)

**iBeacon Prefix:** This is a fixed field, which contains manufacturer specific data, and also information about length of packet.

**UUID:** An identifier in the format B9407F30-F5F8-466E-AFF9-25556B57FE6D. This is commonly used to distinguish all beacons used in a particular context. All beacons in an IPS solution should have the same UUID for example.

**Major number:** Used to group beacons into smaller partitions. An airline company may for example use the same major number for all beacons at Frankfurt Airport.

**Minor number:** Used to identify a single beacon within a group.

**TX Power:** Also known as measured power value. Used for distance estimation. This value can be calibrated by the developer, and be equal to the average RSSI picked up one meter from the beacon. Usually the most accurate value for this field is set as default by the manufacturer.

2.3.2 Eddystone
Eddystone was introduced in July of 2015. It is an open source protocol, which in contrast to iBeacon contains multiple packet types. These contain different kinds of data and can be broadcasted simultaneously by a beacon. Packets contain up to 31 bytes have the following structure:

- **Prefix** (11 bytes)
- **Frame Data** (up to 20 bytes)
The prefix has the same function for all packet types, containing information about the manufacturer and the length of the data. “Frame Data” contains data specific for the packet type. There are currently three packet types: Eddystone-UID is a unique identifier, Eddystone-URL contains a URL address, and Eddystone-TML contains telemetric data.

**Eddystone-UID** (16 bytes): Contains an identifier for the beacon, and is set up as follows:

- **Namespace** (10 bytes)
- **Instance** (6 bytes)

Namespace has a function similar to iBeacons UUID that assigns an identifier to all beacons in a context. Instance is used to group beacons, as well as identify individual beacons.

**Eddystone-URL** (6-20 bytes): This is used to broadcast a website address, and contains a single field as “URL”. The size of the field depends on the length of the URL. With iBeacon and Eddystone-UID there is need for an app to take the identifier of the beacon and translate it into actions, but with Eddystone-URL, the required data is encoded directly into the packet.

**Eddystone-TML** (14 bytes): Telemetry packet, which consists of the following information:

- battery voltage, which can be used to estimate the battery level of a beacon,
- beacon temperature,
- number of packets sent since the beacon was last powered-up or rebooted,
- beacon uptime, i.e., time since last power-up or reboot.

Eddystone-TML is designed to be broadcast alongside another packet, i.e. UID and/or URL. It can be used to notify the owner that, for example, the battery is running out.

### 3. METHODOLOGY

In this work, the primary step is to perform a thorough literature review on the existing algorithms, designed for indoor positioning using BLE Beacons. This will give in-depth knowledge about the optimal placement of beacons, distance algorithms, the common error factors and some intuition on the processes of minimizing error, and algorithms for increasing accuracy. The information obtained from the literature will then be applied in the implementation.

We then select two common commercial solutions, and implement them using a smartphone, and up to five BLE Beacons, placed accurately in different test environments. The solutions will be tested in the same environments, so that a comparison can be made. Common error factors will be specifically tested in some dedicated tests, to see if and how the results are affected.

The test results will then be put together and compared. This will be followed by an analysis of the results, providing intuition behind the performance of each algorithm. Finally, we will conclude the thesis by suggesting and explaining possible improvements.
4. LITERATURE REVIEW
This section will present the different factors affecting localization accuracy, as well as methods to improve accuracy. Algorithms for node placement, distance estimation, noise reduction and localization will be explained.

4.1 Factors affecting positioning accuracy
The accuracy of an IPS is influenced by multiple factors. BLE Beacons are produced by a vast number of third party vendors that vary in quality and signal stability. The transmitting power of the beacon can be adjusted to the optimal (i) power level (see 2.1). On the receiveride, (ii) received power indication is an important factor. The RSSI reading depends upon the hardware of the receiving device. This is why most IPS solutions suggest certain hardware that is best adapted to the software.

The RSSI reading must then be interpreted by the software. This requires an accurate (iii) propagation model (see 4.2) and (iv) calibration of its' variables, such as attenuation factor and antenna gain [20].

(v) Multipath propagation is a major error factor, which implies Bluetooth signals reaching the receiver device by multiple paths. This causes an irregularity in the strength of the signals reaching the device, making the RSSI fluctuate heavily over time [20, 21]. (vi) Obstacles in the environment affects the paths taken by signals and can weaken signals. This has the most obvious effect when the line-of-sight between transmitter and receiver is blocked.

(vii) Lack of signals from at least three beacons, will limit the localization capability considerably. Contact with one beacon can only position us somewhere in the signal radius of that beacon, and two beacons will give us a position in the intersection of their coverage areas. At least three beacons is required for unambiguous positioning.

Fig 4.1 The area of uncertainty is affected by the user’s position as well as beacon placement[18].

(viii) Adverse reference nodes geometry affects the so called area of uncertainty, which covers the possible locations of the user [18]. As shown in Fig 3.1, the user is easier to locate when positioned between the beacons. The area of uncertainty is greatest when the user is positioned outside the triangle, with the beacons in a straight line.
4.2 Node Placement

Maximum Sum of Maximum RSS Model

Optimal placement of nodes in an IPS can greatly improve accuracy. Two studies [9, 10] proposed solutions to improve the accuracy using an algorithm called Maximum Sum of Maximum RSS Model (MSMR) in an IPS using fingerprinting. It is based on Simulated Annealing (SA), a heuristic technique used to approximate global optimization in a large search space.

\[
\text{Maximize } \sum_{i \in T} \max_{j \in R}(S_{ij}P_{ij}) \tag{1}
\]

Constraints:

\[
S_{ij}(P_{ij} - P_T) \geq 0 \quad \forall i \in T, \forall j \in R \tag{2}
\]

\[
\sum_{j \in R} S_{ij} \geq N_R \quad \forall i \in T \tag{3}
\]

\[
S_{ij} \leq c_j \quad \forall i \in T, \forall j \in R \tag{4}
\]

\[
\sum_{j \in R} c_j \geq N_S \tag{5}
\]

**Notations**

Sets:
- \(R\): A set of candidate sites to install reference nodes (RNs).
- \(T\): A set of signal test points.
- \(c_j\): A binary \(\{1, 0\}\) variable that equals 1 if the RN is installed at site \(j, j \in R\); 0 otherwise.
- \(S_{ij}\): A binary \(\{1, 0\}\) variable that equals 1 if the STP is assigned to RN \(j, i \in T\) and \(j \in R\); 0 otherwise.
- \(P_{ij}\): The signal strength that a STP \(i\) receives from RN \(j, i \in T\) and \(j \in R\) (dBm).
- \(P_T\): The received signal strength threshold (dBm).
- \(N_R\): The minimum number of RNs recommended [5].
- \(N_S\): Sufficient number of RNs.
The method uses a set of signal test points (STP) evenly distributed in the test area, from where the RSSI from reference nodes (RN) i.e. beacons is measured. The idea is to maximize the RSSI when summed up from all nodes and signal test points, according to objective function (1) in Fig 4.2. A series of constraints are used to achieve this. Constraint (2) optimizes the quality of the radio signals, by making sure that the signal strength $P_{ij}$ that a STP receives from any RN is above a certain threshold $P_T$. Constraint (3) makes sure that any STP is in range of a certain number of RNs. Constraint (4) specifies that a STP can pick up signals only from a RN $c_j$ which has been installed at the site.

Firstly, a random placement of RNs is selected, and this is set as a current state. Then the SA algorithm will apply the constraints to the current state. A decision function is used to decide whether or not each location will be selected, and if not, a neighbouring location will be selected for the new state. The procedure is then repeated for each new state until optimal placement has been selected, giving us the maximum possible summarized RSSI. Fig 4.3 gives an example of optimal RN placement and results from using the algorithm [9].

**4.3 Methods for distance estimation and noise reduction**

This section describes methods to improve the accuracy of distance estimation. A propagation model is used to predict the loss of signal strength over distance. To reduce the noise a filter must be used. One of the most common ones is the Kalman Filter, which is described in this section. Finally, the importance of window size is discussed.
4.3.1 Propagation model
Propagation loss is the reduction in power density as a radio signal travels through a space. The received signal strength is used to estimate distance. A propagation model is used to transform the RSSI into distance. In [2] the propagation model being used is expressed as:

\[ r = p - 10n \log_{10}(d) \] (6)

where \( r \) is the received signal power at the distance \( d \), \( p \) is the received signal power of the receiver from a transmitter one meter away and \( n \) is again the path loss exponent. The parameters \( p \) and \( n \) must then be calibrated with numerous test; in this particular study [2] they were calibrated to \( p = -70.09 \) and \( n = 1.95 \).

Below is a graph, showing distance estimation using a propagation model over time [14]. The signals are heavily influenced by noise.

![Graph showing distance estimation using a propagation model over time](image)

**Fig 4.4 Distance estimation using a propagation model over time [14]**

4.3.2 Kalman Filter
Even using the propagation algorithm the estimations are not very accurate. Multi-path reflection, meaning that the signals bounce against objects in the environment, is a major factor that influence distance estimation. This causes a lot of noise, making RSSI fluctuate heavily, especially over long distances. To reduce the noise a filter can be used.

The Kalman filter uses the history of measurements to make estimations of the next RSSI. Ideally, we have a constant RSSI when standing still, everything else is noise. A transition model is used to translate the current state to the expected next step:

\[ x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \] (7)
where the state $x_t$ is the combination of the previous state $x_{t-1}$, given a transformation matrix $A$, a control input $u$ and the process noise $\epsilon$.

We also must define how a state $x$ result in a measurement $z$, using the following model:

$$z_t = C_t x_t + \delta_t$$

(8)

where $C$ is the transformation matrix and $\delta$ is measurement noise - noise caused by faulty measurements.

The algorithm involves two steps, prediction and measurement. In the prediction step, the most likely next state is calculated according to our knowledge about previous measurements and noise levels.

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

(9)

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

(10)

When the prediction $\mu$ is made, there is also an estimation of how likely the prediction is to be correct, this is represented by the certainty $\Sigma$. The certainty is based upon how certain the last prediction was, as well as the noise $R_t$.

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

(11)

The Kalman gain $K$, weighs between the certainty of the prediction and the certainty of the measurement. For example, if the measurement has a high certainty, whereas the prediction has low certainty (in the case of lots of noise), the measurement should be trusted more than the prediction. This gives us the update step:

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

(12)

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

(13)

Here the final prediction $\mu$ is made. The larger the Kalman gain the more the measurement is used in the state estimate. The result is a more stable distance estimation, which defines the noise and filters out a large part of it.

4.3.3 Window size

Signal strength can differ greatly between packets, resulting in unreliable distance estimation. One study [11] shows that increasing the window size and evaluating multiple packets simultaneously, results in significantly better estimation. On the other hand, this approach creates larger delays for signal processing and distance estimation, since multiple signals must be received before processing.

Below is the result of tests using different window sizes:
Shows distance estimation error for window sizes 1, 5 and 10 [11]

4.4 Localization techniques

There are three main localization techniques used by BLE Beacons: proximity, trilateration and fingerprinting (FP) [6].

Proximity algorithms are the most basic algorithms that simply indicate location when the mobile device is within a certain coverage area, meaning that the Received Signal Strength from a single beacon is stronger than a pre-set threshold. This report will not go further into this method, but instead focus on the more complex and accurate methods.

Trilateration [7] is a simple range-based localization algorithm using geometry principles. The mobile device collects packets received by beacons, where an application estimates the distance to each beacon. It then calculates its own position by intersecting the circles centered on the beacons’ positions. Ideally, the intersection would be a single point on a surface, but due to different error factors, the intersection identifies an area where the mobile device is likely to be found.

Fingerprinting [8] involves an offline phase and an online phase. In the offline phase, a signal strength map must be created, by recording the signal strength from several beacons along with the coordinates of the client device and storing them in a database. The online phase involves the mobile device receiving RSSI from different beacons, matching the data to the fingerprints in the database, and showing the most similar one as the user’s position. One of the downsides with fingerprinting is that fingerprints degrade over time, as the environment changes - the position of furniture, density of people, and even the positions of walls and partitions [12]. This means that fingerprinting requires continuous resurveying to ensure the accuracy of the system.
A few algorithms will now be explained. 4.4.1 is a common fingerprinting algorithm and 4.4.2 explains a three approaches to one-dimensional positioning.

4.4.1 Fingerprinting using Euclidean Distance Correction Algorithm

A fingerprinting approach is used and described in [13]. The testing area is divided into blocks, by vertical and horizontal lines. At each intersection, the beacons with the strongest RSSI are selected, and their values are stored into a row in a database in a cloud storage. When the user then scans the area, the received RSSI values are compared to each row in the database and the closest match will be shown as the user’s position. The study discusses the possibility of using a trilateration, but because of many obstacles in the environment, fingerprinting was considered to be the most accurate technique.

In the same study, a Euclidean Distance Correction algorithm is used to filter and adapt the results of distance estimations. The three rows in the database which have the shortest Euclidean distance will be calculated and its position will be shown as the user’s position, according to the following algorithm:

\[
d(M, m, N) = \left( \frac{\sum_{i=1}^{N} (M_i - m_i)^2}{N} \right)^{\frac{1}{2}}
\]

(14)

M represents the signal strength stored in the database and m is the current values uploaded by the smartphone. i represents the sequence number of matching groups. N is the number of beacons selected for measurement. The beacons selected are the ones with the strongest RSSI, and the amount of selected beacons depends on need and environment.

One of the issues noted in the article, is that beacons more than 10 meters away, have an RSSI fluctuating slightly between -99 dBm and 101 dBm. The RSSI does not fluctuate enough to give us any useful information about the distance, when the beacon is this far away - so it’s valueless to store it in the database. In fact, these values may lower the calculated Euclidean Distance significantly compared to the actual one, and make the calculation inaccurate. To solve this, the following correction is presented: Whenever the MAC address of a beacon uploaded by the phone that can’t be matched with the row in the database, this means the beacon is very far away. In this case, another parameter of 1024 is added to the Euclidean distance, according to the following algorithm:

\[
d(M, m, N) = \left( \sum_{i=1}^{N} (M_i - m_i) + (10 - N) \times 1024 \right)
\]

(15)

This corrected algorithm, adapts to the beacons which are so far away that the RSSI is no longer related to distance. 1024 was selected as the optimal correction parameter.
4.4.2 One dimensional positioning using three alternative approaches

[15] uses a somewhat simplified approach. The goal is here to locate shoppers in a mall, where shoppers are usually located in aisles between shelves. The user’s position with regard to the width of the aisle is not very important, so the idea is to locate the user on a defined axis parallel to the aisle. The axis is represented in red, and beacons are represented with circles, on the figure:

![Fig 1 Test setup using one-dimensional positioning approach. The circles represent beacons, and the red line represent the axis on which estimated positions are found. [15]](image)

This axis-based approach, simplifies the positioning problem from two-dimensional to one-dimensional. It uses a path loss model similar to ones discussed earlier(6):

\[
s_d = s_0 + 10n \log\left(\frac{d}{d_0}\right) + N(0, \sigma) \tag{16}\]

\(N(0, \sigma)\) in this case stands for Gaussian noise.

Three different approaches for positioning are tested, and the results are compared.

4.4.2.1 Baseline method: Nearest Beacon

The simplest method is to decide which beacon is the closest and show the user’s position as the point closest to the beacon on the axis (The yellow square in Fig 1). Given a series \(R_{b,t}\) of \(k\) readings for a specific beacon \(b\) at a time \(t\) over time window \(w\):

\[
R_{b,t} = \{r_k, r_{k-1}, r_{k-2}, \ldots, r_1\} \tag{17}\]

The study experimented with two different methods for estimating the signal value \(S_{b,t}\) for beacon \(b\) at a time \(t\):

\[
S_{b,t} = ar_k + (1 - \alpha)S_{b,t-1} \tag{18}\]

\[
S_{b,t} = \max_i(r_i \in R_{b,t}) \tag{19}\]

where \(\alpha\) is the learning rate parameter for an exponential moving average. The closest beacon is presumed to be the one with the highest signal value \(S_{b,t}\) at a time \(t\). Different
window sizes were tried, with window \( w_t \) ranging from 1 to 3 seconds, but without much difference in performance.

4.4.2.2 Weighted Beacon-Pair Range Estimates
The study claims trilateration to be unreliable according to [16], and suggests a method where the readings of neighbouring beacons are used in the estimation. First the closest beacon is calculated, then the next closest (which is often close to the first). Given positions \( p_1 \) and \( p_2 \), corresponding distances \( d_1 \) and \( d_2 \) (calculated according to Euclidean Distance Estimation), the target \( p_t \) is calculated:

\[
p_t = p_1 + (p_2 - p_1) \left( \frac{d_1}{d_1 + d_2} \right)
\]  

(19)

4.4.2.3 Particle Filter
The third method uses a Sequential Importance Resampling (SIR) filter [17], which is a particle-filtering algorithm. This is initially a two-dimensional approach, but the state space is restricted to the axis, making it a one-dimensional approach, just like the previous ones.

Particles move along the axis until reaching a node. Then, a new path is selected at random. The next state of the particle is predicted using:

\[
p_{i,t} = v_{i,t} t_s
\]

(20)

Where \( t_s \) is the time since the last update, and \( v_{i,t} \) is velocity. When an RSSI is received, the weight of each particle is computed. The initial position of particle \( p \) corresponds to the beacon position, and the expected signal strength \( s_d \) is calculated according to equation 1. The weight is then defined as the likelihood of the measured RSSI, and calculated by the following algorithm:

\[
w_{i,t} = \frac{1}{\sqrt{2\pi \sigma_p^2}} e^{-\frac{1}{2} \left( \frac{s_m - s_d}{\sigma_p} \right)^2}
\]

(21)

The weight is then normalised, and \( N_p \) new particles are drawn by randomly selecting particles from current distribution. The weight decides the probability of each particle being drawn. The user's position at a time \( t \) is calculated using:

\[
p_u = \sum_{i=1}^{N_p} w_{i,t} p_{i,t}
\]

(22)
4.4.2.4 Results

<table>
<thead>
<tr>
<th>Positioning method</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Beacon Mean(m)</td>
<td>2.15</td>
<td>2.01</td>
<td>1.82</td>
<td>2.01</td>
</tr>
<tr>
<td>Nearest Beacon Std. dev.</td>
<td>1.49</td>
<td>1.47</td>
<td>1.35</td>
<td>1.44</td>
</tr>
<tr>
<td>Beacon-Pair Range Mean(m)</td>
<td>1.53</td>
<td>1.78</td>
<td>2.03</td>
<td>1.76</td>
</tr>
<tr>
<td>Beacon-Pair Range Std. dev.</td>
<td>1.45</td>
<td>1.62</td>
<td>1.72</td>
<td>1.59</td>
</tr>
<tr>
<td>Particle Mean(m)</td>
<td>1.04</td>
<td>1.07</td>
<td>1.44</td>
<td>1.16</td>
</tr>
<tr>
<td>Particle Std. dev.</td>
<td>0.97</td>
<td>1.24</td>
<td>1.31</td>
<td>1.18</td>
</tr>
</tbody>
</table>

The experiments were done in an area of 800 m² using 25 beacons, with a pathway length of 85 m, and data was collected in three separate tests. The table shows the mean error, as well as the standard deviation, from all methods and tests. When comparing the results, Particle Filter proves to be the superior method of the three.
5. EVALUATION

The evaluation used two types of software in a few different scenarios. One small and one larger environment, as well as a hallway environment was tested. Some specific tests were made to evaluate the effect of microwave interference, and to compare a trilateration approach to a fingerprinting approach.

5.1 Setup

This section will describe details of the test setup, both hardware and software. The hardware consisted of four BLE Beacons and an Android phone. The software picked for evaluation were two commercial solutions, GoIndoor and Navigine.

5.1.1 Hardware

The hardware setup consisted of BLE Beacons from Avvel, and an android phone.

5.1.1.1 Avvel Short Range iBeacon

Four Short Range iBeacons from Avvel were used for the experiments. They are a low-cost beacon with a range of up to 30 m, according to the website. The power mode can be set between -20 to +4 dBm. For the experiments the power mode was set to -4 dBm, which was suggested by the GoIndoor company (see 5.1.2.1).

By default, the broadcast rate is set to 1s, one broadcast per second. For the experiments, it was set to 100ms, which significantly lowers the lifespan of the beacon, but gives more frequent updates of the position estimate. 100ms is also the iBeacon standard broadcast rate.

5.1.1.2 Android phone

An android phone, Google Nexus 5, was the device running the positioning software, whose position was estimated in the experiments.

5.1.2 Software

Two commercial solutions, Navigine and GoIndoor, were used in the experiments. They have a similar method of setup and deployment. Maps are uploaded to an online content management system (CMS), where measurements can be made to determine the scale, and paths can be added to create a pathfinding experience. A mobile application is then used recognize and place the beacons on the map.

5.1.2.1 GoIndoor

GoIndoor is a software by the company OnYourMap. It provides an online CMS, which handles routes, multiple floors and trigger zones where notification messages can be triggered. The user uploads a map (blueprint of the building) to the CMS, which is then matched to a real world map similar to Google Maps. By matching the blueprint to the world map the scale is set.

Beacons are added through a mobile application, Quick Setup Tool, by holding the phone near a beacon until it is found and then placing it on the map. There is no calibration option available.

At the time of the evaluation the recommended beacon hardware was kontakt.io, which was disregarded due to long shipping times. GoIndoor now offer their own beacon hardware.
5.1.2.2 Navigine
Navigine is an IPS developed by a company with the same name. It is similar to GoIndoor, and lets the user upload maps to an online CMS, where routes and trigger zones for push notifications can be added. It also allows for walls and barriers to be drawn, which supposedly gives more accurate positioning.

Navigine’s CMS lets the user determine the scale by drawing a line across the map and writing how many meters this represents in real life.

Beacons can be added through the CMS, but also directly in the mobile application. If added in the CMS, Navigine offers a way to calibrate the beacon by setting a “RSSI per meter” value. If added using the mobile application this option is not available. By holding the phone close to the beacon, the beacon is registered and the signal strength is measured.

5.1.3 Test Areas

Fig 5.1 Test areas A and B with measurements. Area A is a hallway, area B is a kitchen with a connected vestibule.
Fig 5.2 Test Area C. Bedroom environment.

The test environment was divided into test areas A, B and C. The majority of the tests in test areas A and B, which are interconnected. They can be viewed with measurements in Fig 5.1 Area A is a hallway containing a staircase, and area B is a kitchen area connected to a vestibule. Test Area C (Fig 5.2) is a smaller environment, a bedroom with measurements 3.6 x 3.4 m.

5.2 Small Area

Fig 5.3 Results. Blue marks beacon positions, red marks the estimated position, black marks the actual position

A small room was used as test environment for the first evaluation, test area C (Fig 5.2). 4 beacons were evenly placed on each of the walls, 2 metres from the ground. The goal was to test the accuracy of trilateration for a small environment, and the ability to pinpoint the user’s position. Results (Fig 5.3) were similar with both Navigine and GoIndoor. No matter the user’s actual position in the room, the estimated position would always point
to the center of the room. However, when in immediate vicinity (30 cm or less) to one of the beacons, the indicator would move from the center of the room near the beacon’s location. To see the relation between RSSI and distance in this environment, the average RSSI from distances 0.5, 1 and 2 meters were tested. Each of the distances was tested from 3 different angles. The measurements in Fig 5.3 show little to no correlation between distance and RSSI when above 50 cm. At 25 cm there is a notable change in signal strength, compared to the other distances. Note that a smaller negative value indicates a stronger signal.

<table>
<thead>
<tr>
<th>Distance</th>
<th>RSSI in the 1st test (dBm)</th>
<th>RSSI in the 2nd test (dBm)</th>
<th>RSSI in the 3rd test (dBm)</th>
<th>Average RSSI (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 cm</td>
<td>-47</td>
<td>-52</td>
<td>-65</td>
<td>-54.7</td>
</tr>
<tr>
<td>50 cm</td>
<td>-69</td>
<td>-63</td>
<td>-87</td>
<td>-73</td>
</tr>
<tr>
<td>1 m</td>
<td>-73</td>
<td>-80</td>
<td>-73</td>
<td>-75.3</td>
</tr>
<tr>
<td>2 m</td>
<td>-73</td>
<td>-72</td>
<td>-78</td>
<td>-75</td>
</tr>
</tbody>
</table>

Fig 5.4 Measured RSSI from 0.25, 0.5, 1, and 2 meters.

5.3 Kitchen environment

This test was made in a slightly larger environment than the last. Two adjacent rooms, a kitchen and a vestibule were used (Test Area B), as well as four beacons. The beacons were first placed in the ceiling, in central positions, as suggested in the MSMR approach[9,10], which optimizes Bluetooth signal coverage. In the second test the beacons were placed on walls. This was done to test how beacon placement affects the results.

Blue dots represent beacon positions, red markers the indicated position, and black the actual position.
5.3.1 Centralized beacon approach

In the ceiling mounted beacon approach, the indicator often shows to the beacon heavy areas, close to one or in between the two beacons in the main room. They are a bit unresponsive, often needing 1-2 seconds before the indicator settles. Sometimes it wanders far off into the walls, multiple metres from the actual position. The hardest positions to estimate seem to be the edges of the map, when the user is positioned far outside the triangle of beacons.
5.3.2 Wall mounted beacon approach

Four beacons were placed on the walls in this test, beacon placement as in Fig 5.7. The idea was to cover up the edges of the map in a better way, by placing beacons in corners of the room instead of central positions. The results are somewhat more accurate, and reaches the edges of the map to a greater extent than earlier. Comparing the two, the centralized beacon approach has an average error of 1.31 m, whereas the wall mounted beacon approach has an average of 0.92 m.
5.3.3 Microwave test

The two solutions were also tested with a microwave running in the kitchen. These tests should be compared with Fig 5.5 and 5.6, which have the same beacon setup. There is no noticeable change in accuracy with the microwave running, and neither in responsiveness. The beacons transmit 10 packets per second, so issues with responsiveness due to packet loss may have gone unnoticed in the tests. The average error from the tests is 1.24, but more tests would be required to completely rule out the effect of microwaves. One beacon was placed near the microwave and tested individually, to see if RSSI was affected, from a distance of 0.5 and 2 meters separately. There was no noteworthy difference.

<table>
<thead>
<tr>
<th></th>
<th>Average error from test 1 (m)</th>
<th>Average error from test 2 (m)</th>
<th>Total average error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoIndoor Centralized Beacons</td>
<td>1.10</td>
<td>1.29</td>
<td>1.20</td>
</tr>
<tr>
<td>Navigine Centralized Beacons</td>
<td>1.22</td>
<td>1.45</td>
<td>1.32</td>
</tr>
<tr>
<td>GoIndoor Microwave Test</td>
<td>1.34</td>
<td>1.46</td>
<td>1.40</td>
</tr>
<tr>
<td>Navigine Microwave Test</td>
<td>1.02</td>
<td>1.14</td>
<td>1.08</td>
</tr>
<tr>
<td>GoIndoor Wall Mounted Beacons</td>
<td>0.55</td>
<td>0.76</td>
<td>0.66</td>
</tr>
<tr>
<td>Navigine Wall Mounted Beacons</td>
<td>0.92</td>
<td>1.03</td>
<td>0.98</td>
</tr>
<tr>
<td><strong>GoIndoor Total Average</strong></td>
<td><strong>0.92</strong></td>
<td><strong>1.03</strong></td>
<td><strong>1.09</strong></td>
</tr>
<tr>
<td><strong>Navigine Total Average</strong></td>
<td></td>
<td></td>
<td><strong>1.02</strong></td>
</tr>
</tbody>
</table>
5.4 Hallway
Four beacons were placed in a hallway, and to adjacent rooms, test area A and B. They were placed in positions suggested by the GoIndoor company, quite centralized in the ceiling. The first test is an accuracy test (trilateration) from both solutions, the second uses the fingerprinting approach provided by Navigine.

5.4.1 Trilateration

Fig 5.11 Trilateration approach by Navigine

The trilateration approach by Navigine had an average error of 1.31 m, and GoIndoor 1.63. The same tendencies as previously were noted - the indicator stays close to where the beacons are placed, and the worst accuracy is registered when beacon light areas. The most extreme scenario was when standing in a doorway between two rooms; when facing room A, the indicator would point to the center of room A, and when turning to face room B, the indicator would move about 4 meters, to the center of room B.
5.4.2 Fingerprinting

![Fingerprinting diagram](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Average error from test 1 (m)</th>
<th>Average error from test 2 (m)</th>
<th>Total average error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoIndoor</td>
<td>1.54</td>
<td>1.72</td>
<td>1.63</td>
</tr>
<tr>
<td>Navigine Trilateration</td>
<td>1.25</td>
<td>1.36</td>
<td>1.31</td>
</tr>
<tr>
<td>Navigine Fingerprinting</td>
<td>1.67</td>
<td>1.61</td>
<td>1.64</td>
</tr>
</tbody>
</table>

Fig 5.13 Accuracy in metres from hallway tests

The same beacon placement was used as in 5.11 for the fingerprinting. Fingerprints were stored in the database, about one for every meter square meter of the test area. The results from this approach were actually worse - the indicator tended even more than in the trilateration approach, to point to the area around the closest beacon. The trilateration approach from Navigine gave the best results from the hallway tests. However, it is hard to rely on any of the solutions for more than room level accuracy.
6. CONCLUSION

This thesis has evaluated the effectiveness of BLE Beacon Technology for positioning. BLE Beacons is a low-cost, low-power solution, and if the technology can be used for positioning in an efficient and accurate way this could be beneficial in many areas, and this has been the motivation for the work. Multiple error factors make the localization process difficult, and the common factors were identified in the literature review. Some existing algorithms which can be used to moderate the error and improve accuracy were then presented and explained. Two commercial solutions were selected and used for evaluation.

The evaluation was done in one small and one larger environment, as well as a hallway environment. The small environment, 3.4 x 3.6 m, turned out too small for positioning. RSSI was measured from different positions in the room, and it was concluded that RSSI had little relation to distance at distances of 0.5-2.0 meters. This made positioning impossible in this small environment.

In larger areas the positioning was more successful. The positioning had an average error between 0.9-1.7 m. The wall mounted beacon approach was slightly more accurate, and covered up the far edges of the map more effectively. Microwaves seemed to have no effect on either accuracy or responsiveness, based on the tests.

A scenario using four beacons had better accuracy than one using three beacons. In some scenarios however, more beacons would be a disadvantage. Beacons that are over 10 meters away have less correlation between RSSI and distance, meaning that picking up a signal from a distant beacon can put you off, rather than contribute to the accuracy. Thereby it is in many scenarios beneficial to be out of range from distant beacon, rather than picking up its misleading information. It was clear in the testing that beacon placement was very important to the results.

The evaluation of Navigine’s two positioning methods, showed better results from trilateration than fingerprinting.

The battery of the phone was drained quite quickly during the tests, so using Bluetooth based positioning for extensive periods of time is potentially an issue to consider. No investigation was made into how much battery life was consumed by the software itself, versus the device retrieving signals and Bluetooth simply being turned on for long time periods.

6.1 Suggested improvements

An improvement for both solutions would be a better and more apparent way to calibrate the software. Navigine offers a RSSI per meter setting in the CMS, but synchronization issues between the application and the CMS made calibration in this way almost impossible. Besides, RSSI per meter is hard to interpret, since the change in RSSI per meter is higher when close to the beacon, and almost non-existent from over 10 meters of distance. The values being manipulated should be the ones in the propagation model, the RSSI one meter from the beacon and the path loss exponent. RSSI per meter could be referring to the path loss exponent, this is somewhat unclear. If the beacon are added using the mobile application no manual calibration is available. Instead the signal strength is measured by holding the phone close to the beacon. The accuracy of the
measurement is questionable, since this is dependent on how close the user is holding the phone. It is hard to believe all necessary exponents can be calibrated this way.

GoIndoor did not offer any calibration options in their solution. It was however clear through email correspondence with the company that their solution uses the measured power value (see 2.3.1) which can be calibrated in the beacon hardware.

Both software solutions encountered some technical issues, but most of them were contributed by GoIndoor. The online map administration was often non-responsive, and not reliable. This would sometimes cause the tests to be postponed another day. GoIndoor had the most features, which would have made it the most complete IPS if it weren’t for the bugs. The scale of the map is obviously an important part in positioning; GoIndoor would let you set the scale by matching the uploaded map to a real world map (see 5.1.2.2). This way of setting the scale was inconvenient, made the image small and blurry, and made the scaling less accurate than it had to be. This could be improved by letting the user set the scale manually, by dragging a line on the map and writing how far that distance represents in real life.
REFERENCES


