UNIVERSITY OF BERN

BACHELOR THESIS

Received signal strength-inertial sensor based fingerprinting localization in indoor wireless environments

Author:
Carl BALMER

Supervisor:
Jose CARERRA

Head of Research

PROFESSOR DR. TORSTEN BRAUN

Communication and Distributed Systems
Institute of Computer Science

May 23, 2017

UNIVERSITY OF BERN

Faculty of Science
Institute of Computer Science

Bachelor of Science in Computer Science

Received signal strength-inertial sensor based fingerprinting localization in indoor wireless environments

by Carl BALMER

Abstract

In this work a new indoor localization system is proposed, combining fingerprint and range-based approaches. We propose a machine learning approach for room recognition and a range-based weighted trilateration method for localization. The weights for the trilateration are defined by considering information provided by the room recognition approach.

We test our localization approach in a complex indoor scenario on the third floor of the Institute of Computer Science at the University of Bern. Results show that our room recognition approach achieves accuracies of 80-90%. Moreover, our weighting method improves the localization by 12% over non-weighted trilateration. It does, however, not perform better than existing simpler weighting methods.

Acknowledgements

I would like to thank Jose Carrera for supervising this thesis and his support on various occasions during my work on this thesis.

Further thanks go to Viviane Tanner and Martina Föhn for their individual support on this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

AN	**A**nchor **N**ode
MN	**M**obile **N**ode
CDS	**C**ommunication and **D**istributed **S**ystems
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
OLS	**O**rdinary **L**east **S**quares
WLS	**W**eighted **L**east **S**quares
LOS	**L**ine **o**f **S**ight
NLOS	**N**on **L**ine **o**f **S**ight
PDR	**P**edestrian **D**ead **R**eckoning
SVM	**S**upport **V**ector **M**achine
RP	**R**eference **P**oint
TP	**T**est **P**oint
k-NN	**K**-**N**earest **N**eighbor
NLR	**N**on-**L**inear **R**egression
CDF	**C**umulative **D**istribution **F**unction
STD	**S**tandard **D**ev**i**ation

Chapter 1

Introduction

Today's ubiquity of mobile computing has increased the demand for mobile devices to be aware of their location. Indoor location awareness is fundamental for many possible applications such as pedestrian navigation and location based marketing in large building complexes (e.g. universities, airports, hospitals).

In contrast to outdoor localization, where the Global Positioning System (GPS) is the most attractive and effective technology to perform object localization, there's no established solution for indoor localization. GPS can't be applied in indoor scenarios due to the inability of GPS signal to penetrate in-building materials such as walls. Moreover, radio localization approaches in indoor environments are affected by non-line-of-sight (NLOS) and multi-path propagation. These effects deteriorate the signal to be less accurate for localization [4, 16]. In addition to these challenges indoor location based applications usually require higher accuracy than those outdoors; An error of four meters is acceptable for street navigation but not for a museum guide.

1.1 Motivation

Indoor localization has been an active research field in the last few years with many different techniques proposed [15, 10]. One common approach is to base the localization on WiFi radio signals. WiFi infrastructure is already present in almost every building and can easily be upgraded with standardized off the shelf hardware. These radio-based techniques are usually classified into range-based and range-free methods [13].

Fingerprinting is a common range-free method, where known radio parameters are mapped to a location. Later this map is used to determine the devices location based on the current radio parameters. Fingerprinting can achieve good accuracy but creating the map is very labor intensive [13].

Range-based methods use the radio parameters to try to approximate the distance between the mobile device (Mobile Node) and the signal emitters (Anchor Nodes). This process is called ranging. Trilateration is then performed on these distances to determine the position of the MN. The ranging process is prone to errors caused by NLOS and multi-path propagation [13].

1.2 Contributions

In this work we propose a localization system which uses fingerprinting to improve the range-based approach.

The range-based localization method is complemented by a room recognition technique based on WiFi received signal strength indicator (RSSI) and magnetic field readings. The information provided by the room recognition is used to define a weighting model, which assign weights to the ranges in the trilateration algorithm.

The proposed localization system is implemented and tested in a complex indoor scenario.

The main contributions are summarized as follows:

- We present a simple, easy to train room recognition method based on fingerprinting and utilizing magnetic field and RSSI information.
- We present novel weighting method for range based trilateration which estimates the ranging error based on the information provided by the room recognition.
- We combine the two above mentioned methods to create an improved indoor location system.

1.3 Overview

The remainder of this work is structured as follows; The theoretical background is reviewed in chapter 2. The proposed localization system is introduced in chapter 3 and the room recognition and weighting are explained in detail. Chapters 4 and 5 deal with the test bed implementation used for the evaluation and the evaluation results respectively. Final chapter 6 summarizes the work and concludes the evaluation results.

Chapter 2

Theoretical Background

In this section two of the most common indoor localization approaches - fingerprinting and range-based are reviewed. This knowledge is required to fully understand our implementation of these mechanisms. Additionally, we also include some information specific to our room recognition approach.

This chapter is organized as follows: First the range-based localization approach is introduced; the important aspects of the observation parameters (RSSI, signal propagation) and each phase of the process (ranging, weighting, trilateration) are covered in detail. Second an overview over the basic fingerprint based approach is given. Finally we cover some topics specific to our room recognition approach; earth's magnetic field and support vector machine (SVM) classification.

2.1 Range-Based Localization

A range based localization system consists of two main components [15]:

Several Anchor Nodes (ANs) which are placed at known locations and constantly transmit a radio signal.

A mobile node (MN), in this case a smartphone, whose location is unknown and needs to be determined.

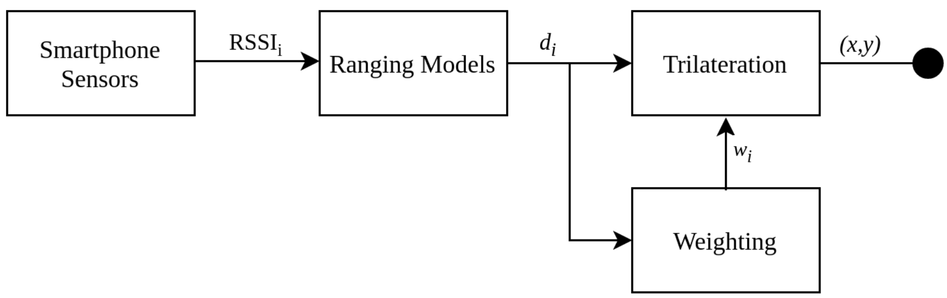

FIGURE 2.1: Block diagram of the range based localization approach.

To determine its position the mobile node measures the received signal strength from each of the anchor nodes ($RSSI_i$). A ranging model is then used to estimate the distance (d_i) from the mobile node to each anchor node. Because the location of the anchor nodes is known, it is then possible to calculate the position of the mobile node using trilateration. To account for errors during the ranging step the trilateration can also be provided with a set of weights (w_i) representing the accuracy of each distance estimation.

The benefit of this approach is that it's not labor intensive. Compared to other approaches only a small number of training samples are required. The disadvantage is that the achievable accuracy is limited. Because of the signal propagation effects in indoor environments the ranging models are often inaccurate.

In the following subsections the ranging, trilateration and weighting steps are described in further detail.

2.1.1 RSSI and signal propagation

The received signal strength indicator describes the signal power level received by the receive radio. The measurement is given in arbitrary discrete units with higher numbers relating to a stronger signal [21].

In an open space without any obstacles the RSSI mainly depends on the propagation distance, but indoors several other factors become important. These are non line of sight (NLOS) and multi-path propagation.

NLOS occurs when the signals path is obstructed by physical objects. The signal must pass through these objects and therefore the RSSI is lower compared to LOS, where there are no obstacles [4].

Multi-path propagation is caused when the signal is reflected from physical objects and arrives at the receiver multiple times with different signal strength. This causes inaccuracy and fluctuations in the measured RSSI as all these signals are blended together [16].

Both effects are very common in indoor environments, caused by the walls, people, furniture and other building materials. Furthermore, the RSSI values are discrete and not fine grained what causes additional inaccuracy. This makes range-based localization based on RSSI challenging and limits its accuracy.

There are other ways to assess the signal strength, such as channel state information, which is more fine-grained and can mitigate multi-path effects, but they are not available on most mobile devices [4, 13].

2.1.2 Radio-based ranging process

The ranging process estimates the distance between the ANs and the MN based on the radio parameters, in this case RSSI. There are several different models that can be used for ranging. In this work we use a non-linear regression (NLR) model proposed by [14]:

$$d_i = \alpha_i e^{\beta_i RSSI_i} \tag{2.1}$$

It describes the loss of signal strength over the propagation distance. d_i is the estimated distance from the MN to the i-th AN, $RSSI_i$ is the i-th AN's signal strength as measured by the MN and α_i, β_i are environment variables specific to each AN.

The model needs to be trained for each AN individually by determining the values for α_i and β_i. This is done by fitting the function to a small set of training samples. This can be done using, for example, least squares optimization.

2.1.3 Trilateration

Trilateration is the process of determining an absolute or relative location based on the distance to known locations. In contrast to triangulation it relies on distances instead of angles.

In the context of localization the goal is to determine the NM's location (x, y) based on the locations of the ANs $(\tilde{x_i}, \tilde{y_i})$ and the distance estimations d_i obtained from the path loss model.

The actual distance D_i from the MN to the i-th AN can be expressed as follows:

$$D_i = \sqrt{(\tilde{x_i} - x)^2 + (\tilde{y_i} - y)^2} \tag{2.2}$$

Under the assumption that $d_i = D_i$ this leads to the following equation system:

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} \sqrt{(\tilde{x_1} - x)^2 + (\tilde{y_1} - y)^2} \\ \sqrt{(\tilde{x_2} - x)^2 + (\tilde{y_2} - y)^2} \\ \vdots \\ \sqrt{(\tilde{x_n} - x)^2 + (\tilde{y_n} - y)^2} \end{pmatrix} \tag{2.3}$$

But d_i is only an estimation so there is no exact solution of the above system. The best solution is the one that minimizes the sum of the squared error $d_i - D_i$. So to determine the MN's location the following problem has to be solved:

$$argmin_{x,y} \sum_{i=1}^{n} w_i \left(d_i - \sqrt{(\tilde{x_i} - x)^2 + (\tilde{y_i} - y)^2} \right)^2 \tag{2.4}$$

To solve non-linear least squares problems the *Levenberg–Marquardt* and *Gauss-Newton* algorithm can be used [19, 20].

2.1.4 Range weighting process

The optimization problem in equation 2.4 also defines a set of weights w_i corresponding to each distance estimation d_i. In the context of trilateration these weights represent how accurate each distance estimation is.

The ranging model's accuracy can vary greatly. By applying a large weight to the more accurate estimations and a small weight to the inaccurate ones it should, in theory, be possible to correct for the ranging error and improve the localization.

In practice the problem is that the ranging error is not known. A weighting method is needed that estimates the ranging error.

Previous work at the CDS group [13] used the assumption that the ranging error is larger with increasing distance to the anchor node. So the weights were defined as inversely proportional to the estimated distances:

$$w_i = \frac{{d_i}^{-1}}{\sum_{n=1}^{N} {d_n}^{-1}} \tag{2.5}$$

In the remainder of this work this weighting method will be referred to as *Distance Weights*.

2.2 Fingerprinting-based localization

Fingerprinting is a common method for localization based on RSSI [6]. It consists of two main phases:

During the **offline/training phase**, a survey of the area of interest is performed. A map of reference points (RP) is created. Each reference point represents a known location and contains the RSSI for each AN.

Then during the **online phase**, a location positioning technique uses the currently observed signal strengths and previously collected information to figure out an estimated location. The positioning technique can employ different machine learning schemes such as k-nearest neighbor regression or support vector machines [4, 15].

The accuracy of this method mainly depends on the density of the RP-map. A higher density usually results in a better accuracy. Generally achieving a satisfying level of accuracy requires a lot of RPs. Other factors are the number of attributes in each RP and the variability of the observation parameters [12].

More attributes per RP, an attribute being a data value like a RSSI or a magnetic field measurement, gives the algorithm more information to work with and so increased the accuracy[12]. This effect is subject to diminishing returns[3]. A high variability in the observation parameters depending on location is also beneficial.

This approach is able to achieve good accuracies in indoor environments. The problem is that it is very labor intensive to create the necessary fingerprinting maps.

2.3 Earth's magnetic field in indoor environments

Earth's magnetic field is the magnetic field that extends from the Earth's interior out into space. It is similar to a magnetic dipole with field-lines pointing towards the magnetic north [18]. This feature has already been used for outdoor localization, mainly as a compass to determine the devices heading in PDR systems.

However, in indoor environments earth's magnetic field is disrupted. The presence of metal structures in the building materials, electrical devices, cables and tubes cause anomalies in the magnetic field. These anomalies make accurate heading determination difficult [1].

But previous research suggests that they can be used in a fingerprinting approach to determine a devices location. The idea is that the presence of a magnetic field anomaly can be linked to a specific location. The research shows that the magnetic field anomalies are mostly stable over time and have sufficient local variability. Therefore, they should be applicable for use in localization [11, 2, 12].

2.4 Support Vector Machine

The Support Vector Machine (SVM) is one of the most widely used machine learning algorithms. It predicts the labels of new (unknown) samples based on previous (known) examples. In its basic form it only supports two labels. This is called binary classification.

The known examples are called training data. It consists of instance-label pairs $(x_i, y_i), i = 1, ..., l$ where $x_i \in R^n$ and $y_i \in \{-1, 1\}$. x_i represents the sample's observable features while the label y_i defines in which category it belongs.

The SVM maps the samples into n-dimensional space. It then tries to fit a hyperplane through that space separating the two classes. Ideally all samples with $y_i = 1$ are on one side of the hyperplane and $y_i = -1$ on the other. To make the separation as clear as possible the margin between the hyperplane and the samples is maximized at the same time. The samples which lie directly on the margins are called the support vectors.

To classify a new unknown sample the SVM determines on which side of the hyperplane it lies and assigns the according label.

To fit the hyperplane the SVM solves the following optimization problem [5]:

$$\begin{aligned} &\min_{\omega,b,\xi} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \\ \text{subject to } & y_i\left(\omega^T\phi(x_i) + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, ..., l \end{aligned} \tag{2.6}$$

There may be outliers or noise in the data. This means that a hyperplane that separates all samples correctly may not be the best classifier. To account for this a cost is paid if a sample violates the error term $y_i\left(w^T\phi(x_i) + b\right) \geq 1$, increasing the objective function by $C\xi_i$. The C parameter defines the trade-off between the simplicity of the decision surface (*hyperplane*) and misclassification of training samples. For large values of C the optimization will chose a hyperplane with a smaller margin and more support vectors. Therefore the *hyperplane* will be more complex, as it tries to classify all samples correctly. Conversely, a small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies some samples [7].

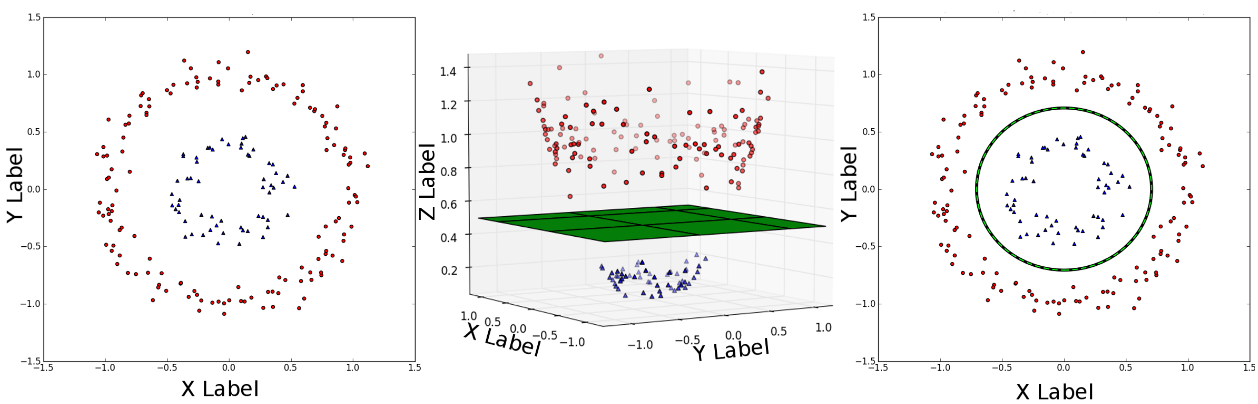

FIGURE 2.2: (Left) A dataset in feature space, not linearly separable. (Middle)The same dataset transformed with decision boundary. (Right) The nonlinear decision boundary.

But the training data may not be linearly separable. In this case the so-called kernel trick can be used. The kernel is a function $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ which maps the features x_i to a higher dimensional space where they can be separated by a hyperplane. This results in a non-linear separation in the original feature space [8].

To perform multi-class classification the "one-against-one" approach can be used. For k classes $k(k-1)/2$ classifiers are trained. Each binary classifier is then considered to vote for a class. The sample is then placed in the class with the most votes [5].

Chapter 3

Localization System Architecture

In chapter 2 the two most common indoor localization approaches were introduced. In this chapter a localization system is proposed that combines those approaches. Our proposal uses a fingerprinting based room recognition system to improve the weights for the range-based trilateration.

After giving an overview of the proposed system we will focus on the new components of our system; the room recognition system and room based weighting method. The other components of the system were already covered in chapter 2.1.

3.1 Overview

The proposed system adds a room recognition system and new weighting method to the standard range-based approach (see chapter 2.1). The room recognition uses fingerprinting with $RSSI$ and magnetic field data to determine the devices current room. The new weighting method then relies on the information provided by the room recognition to more accurately estimate the ranging error and improve the trilateration accuracy.

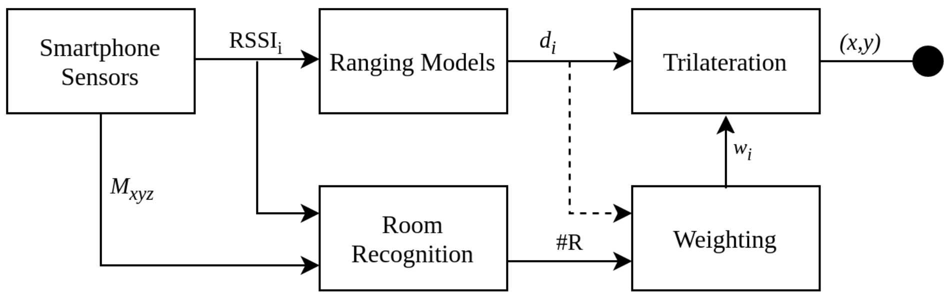

FIGURE 3.1: Block diagram of the proposed localization system.

As apparent in the block diagram (Figure 3.1) the standard range-based approach is not changed but simply extended. Therefore, the main focus of this thesis are the two added components; the proposed room recognition system and weighting method for the trilateration. In the remainder of this chapter those components are explained in detail.

3.2 Room Recognition

The room recognition system is based on a fingerprinting approach. The fingerprint-map consists of $RSSI$ and magnetic field data (B_{xyz}). A multi-class SVM classifier is trained with the fingerprinting map and can then be employed to predict the devices room.

Fingerprinting can be very labor intensive when used for accurate localization, because for each of the many training samples the exact location needs to be measured. But for room recognition we don't need to know the exact location of each sample, we only need to label them with a room. This means collecting the samples to train the room recognition should be a lot faster.

Magnetic field data (B_{xyz}) is also included in the fingerprinting map. We predict that this will increase the accuracy of the room recognition system as previous work has shown the applicability of magnetic field data for localization. *This hypothesis needs to be confirmed in the evaluation.*

Another question to be answered is what kind of fingerprinting map yields the highest accuracy; *an equally distributed map or a unequally distributed one with more samples at the borders* (walls and doors between rooms). Or in other words; do more samples at the borders increase the accuracy.

The support vector machine was chosen as the classifier because it is, compared to other common classifiers like k-NN, better suited for this kind of problem. The SVM is pretty resistant to outliers in the training data, because it only chooses the most significant samples as support vectors. Also, it performs well with a small number of samples. k-NN on the other hand is very susceptible to outliers and generally needs more samples to offer good results.

3.3 Room based weighting method

In order to be able to set trilateration weights based on the rooms, we need a model which relates the rooms to the ranging error.

If we assume that the ranging error is mainly caused by obstructions to the signal (walls, wires etc.) then for any two samples inside one room the ranging error should be roughly the same. Therefore, it should be possible to estimate the ranging error of an unknown sample by calculating the average error of some known training samples located in the same room.

The proposed weighting method defines the weights for each room as inversely proportional to the average ranging error for each anchor node. The result is a separate set of weights for each room. The room recognition is used to decide which set of weights to use.

For room R the weight w_{Ri} associated with the distance estimation d_i to anchor node AN_i is calculated based on all the training samples S_R in the room.

$$\begin{aligned} w_{Ri} &= \frac{E_i^{-1}}{\sum_{n=1}^{N} E_n^{-1}} \\ E_i &= \sum_{s=1}^{S_R} (D_{si} - d_{si})^2 \end{aligned} \tag{3.1}$$

The inverse ranging error for anchor node AN_i is divided by the sum of the inverse ranging error for all anchor nodes N. The ranging error is represented by the sum of the squared difference between the actual and estimated distance $(D_{si} - d_{si})^2$ for each training sample s $(s = 1..S)$.

In the remainder of this work this weighting method will be referred to as *Room Weights*.

I could also be beneficial to combine the new *Room Weights* with the *Distance Weights* from equation 2.5 by adding them together.

The effectiveness of both the *Room Weights* and *Room+Distance Weights* will need to be evaluated.

Chapter 4

Localization System Implementation

In order to develop and evaluate the system outlined in Chapter 3, it is first required to establish a testing environment. This chapter introduces the implementation of the localization system in a test bed.

4.1 System Overview

The test bed consists of multiple anchor nodes, a mobile node and a computer.

The ANs are commercial WiFi access pints, which are placed in the area of interest and constantly broadcast a beacon signal.

The MN is an Android smartphone. It is used to collect samples from different locations in the area of interest.

The samples collected by the MN are transferred to a computer. The computer is responsible for all the computations. It executes all the algorithms for the room recognition, ranging, weighting and the trilateration.

FIGURE 4.1: Overview of the test bed implementation

In our test bed implementation the smartphone is only used to collect the samples. On the computer these samples are then used as training and evaluation points for the localization system. This way it is possible to try out and empirically compare different parameters of the system under the exact same conditions.

The set-up of this test bed comprises both hardware and software specific configurations for each component. The remainder of this chapter details these configurations.

4.2 Hardware

The hardware for this test bed consists of multiple ANs, a MN and a computer. There are no special requirements for the computer as long as it is able to execute Java code.

For the AN and MN the following hardware was employed:

Anchor Nodes The commercial WiFi access points used as anchor nodes are of the model D-Link D-635 and D-2553. They are set-up with a beacon period of 100ms and broadcast on the 2.4 GHz frequency band.

Mobile Node The mobile node is an Android smartphone of the model *One Plus One*. It has the following specifications:

- **OS:** Android 5.1
- **Processor:** 2.5GHz Quad-core CPU
- **WiFi module:** Qualcomm WCN3680 802.11ac/FM/BT 4.0 Combo Chip
- **Internal sensors:** accelerometer, magnetometer, gyroscope, proximity, ambient light
- **Memory:** 3 GB RAM

The WiFi module and the magnetometer are used for the sample collection. The magnetometer is reasonably accurate with a resolution of 0.1 µ T. The off-the-shelf WiFi interface is less reliable because it does only provide fine-grained RSSI values (more fine-grained values would be better) and it does not support channel state information.

4.3 Software

4.3.1 Mobile Node

The mobile node runs an Android application which collects RSSI and magnetic field data.

To collect one sample the application takes the average of five RSSI and magnetometer measurements, each spaced 2 seconds apart. The samples are then saved to a `.csv` file on the smartphones internal storage so they can later be transferred to the computer.

Each sample consist of:

- **A label** indicating the room or the exact location where the sample was taken.
- **A set of RSSI values**, one for each AN.

- **The magnetic field strength** in μ-Tesla along the devices x,y and z axis.

On Android the WiFi module cannot be accessed directly. WiFi scans have to be initiated through the `AndroidAPI` and it only supports full scans [3]. Full scans take longer so it is only possible to take one RSSI measurement every 1.5 seconds.

Due to the low sampling rate, it is not practical to apply filters to remove noise from the $RSSI$. It is also not possible to access channel state information which could be used to mitigate some of the multi path effects.

4.3.2 Computer

On the computer a few different programs are used to implement the localization system:

- **WEKA 3.6** is used for the room recognition classifier.
- **Matlab**'s function fit tool is used to train the ranging model.
- A **Spreadsheet** program is used to manually calculate the weighting model.
- The **trilateration tool** [9] is responsible for the application of the ranging and weighting models and performs the trilateration. It is a small Java application written for this test bed implementation.

The implementation of the room recognition system is split into two phases; In the **offline phase** the room recognition, ranging and weighting model are generated. Afterwards, in the **online phase**, these models are used to predict the location of unknown samples.

Figure 4.2 gives an overview of this process. In the following the two phases are explained in detail.

Offline Phase

Room Recognition Model The SVM for the room recognition is trained in WEKA. It generates a multiclass SVM model from the training data set and may use grid search for the parameter selection. The training data set consists of samples containing RSSI ($RSSI_i$) and magnetic field (B_{xyz}) values labeled with the room number ($R\#$).

Ranging Model For the ranging model the α, β parameters from equation 2.1 need to be determined for each AN. This is done by fitting the equation to the testing data in *Matlab*. The testing data set is a list of RSSI values and the corresponding distance (D_i) to the AN.

The resulting non-linear regression model is inaccurate with high RSSI values (samples very close to the AN). To account for that the distances for these high values are set by hand.

Weighting Model The *Room Weights* are calculated by hand based on equation 3.1 in a spreadsheet program and imported into the trilateration tool.

The *Distance Weights* are calculated by the trilateration tool during run time based on the equation 2.5.

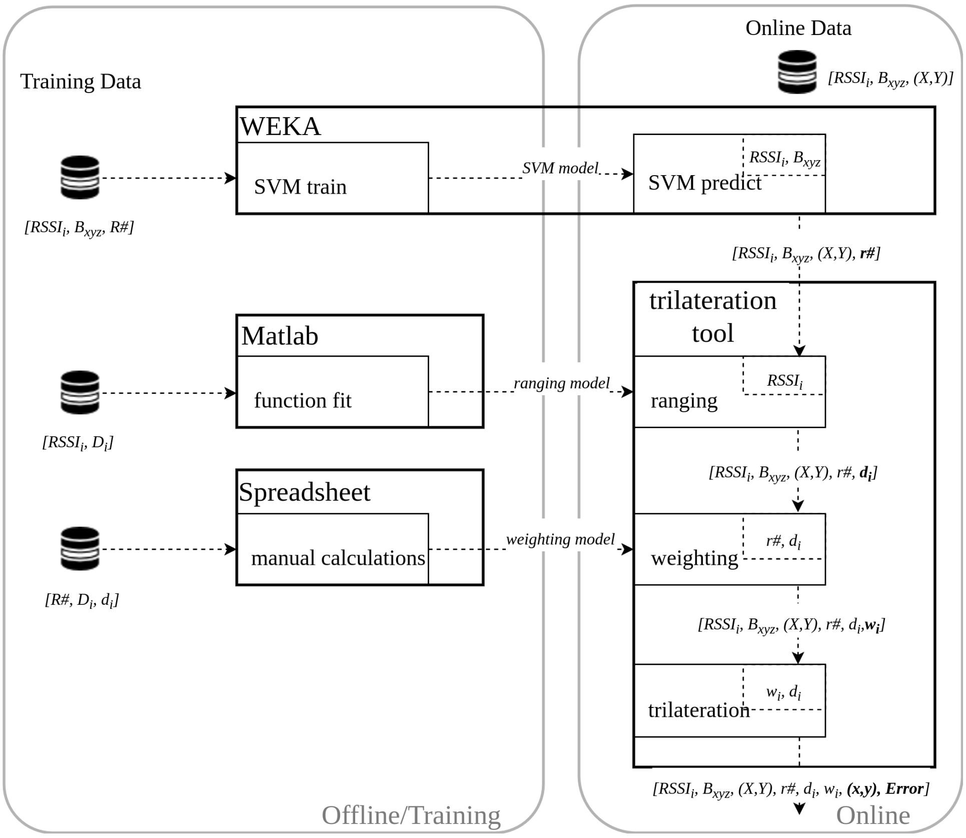

FIGURE 4.2: Diagram of the offline implementation

Online Phase The online dataset contains samples with RSSI and magnetic field values.

In a first step WEKA is used to predict the room number. The result is handed to the trilateration tool, which applies the ranging and weighting models to determine the predicted distances (d_i) and calculate the weights (w_i). It then solves the trilateration problem (equation 2.4) using the Levenberg–Marquardt optimizer from the `Apache Commons Math` library. It outputs the predicted position (x, y) to a `.csv` file.

To evaluate the system, the online dataset can also contain the actual position (X, Y) where the sample was collected. In this case the trilateration tool will also output the localization error.

Chapter 5

Evaluation

In chapter 4 the test bed implementation was introduced. In this chapter this test bed is used to evaluate the performance of the system proposed in chapter 3.

The evaluated components of the system are the room recognition and the room based weighting method for the trilateration.

Before presenting the evaluation results section 5.1 introduces the environment where the test bed was deployed and the data sets used for the evaluation. Sections 5.2 and 5.3 then evaluate the room recognition and weighting method.

5.1 Test bed deployment and collected data sets

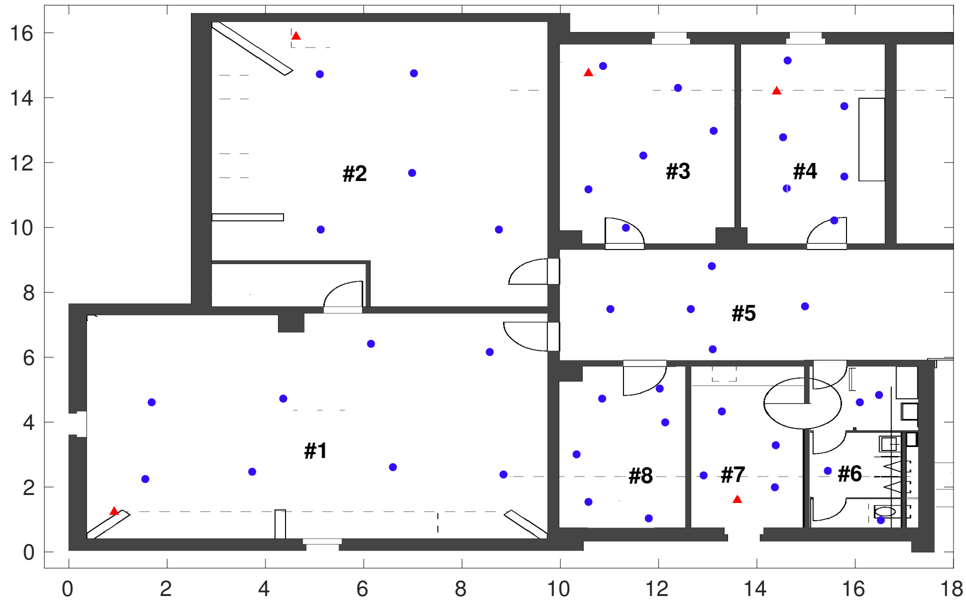

FIGURE 5.1: Floor plan showing the area of interest with the anchor nodes in red, the *XY*-samples in blue and the room numbers in black

The test bed was deployed on the third floor at the Institute of Computer Science (INF) of the University of Bern. The area of interest is 297m^2 in size, with seven rooms connected by a large corridor. The five anchor nodes were positioned to provide maximum coverage of the area so that the mobile node is able to receive at least four of the signals at all time.

The rooms were given numbers from one to eight, the corridor being also treated as a room.

Collected Data Sets The samples were collected with the smartphone held approximatively one meter above the ground and always pointing in the same cardinal direction. This is important because the magnetic field measurements are influenced by the devices orientation.

The collected samples were grouped into the following data sets:

- **Room recognition data only** labeled with the room number
  - Grid (223 samples)
    A set of **evenly distributed** samples gathered in a grid pattern with approximately 1.2m distance between them.
  - Borders (373 samples)
    A set of **unevenly distributed** samples. The sample density is very high at the borders (walls and doors between rooms) with about one sample taken every 0.5m but only a few samples from the center of each room.
- **XY data** labeled with the exact coordinates
  A set of 44 **evenly distributed** samples labeled with (X, Y)-coordinates **and** room number. Figure 5.1 shows them as blue dots.

The room recognition data sets (*Grid*, *Borders*) are used to train the room recognition model. The *XY* data set is mainly used to train the ranging model and evaluate the trilateration accuracy. But we can also use it as an independent test set to compare and evaluate the room recognition models.

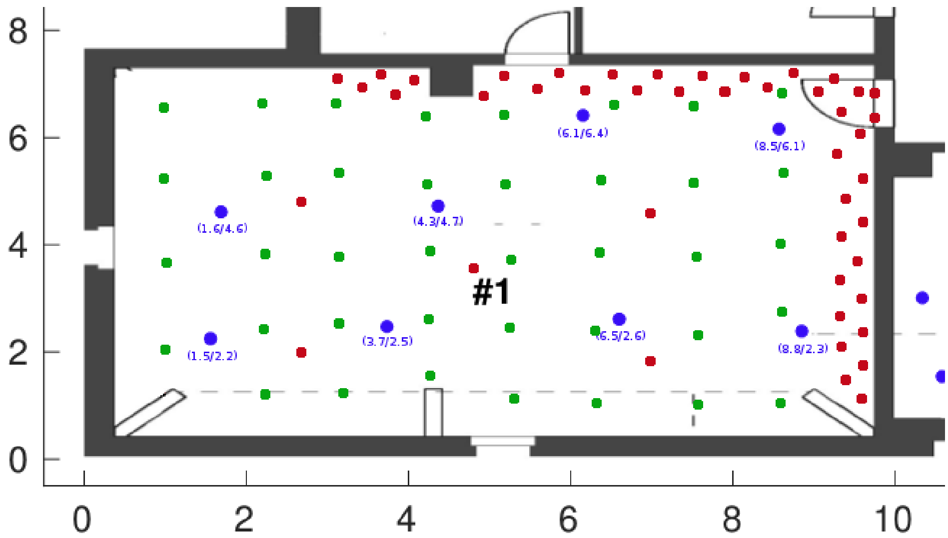

FIGURE 5.2: Illustration showing the collected data sets. (blue) *XY* data set with exact coordinates, (green) *Grid* data set, (red)*Borders* data set.

Evaluation Type	with B_{xyz}	without B_{xyz}	Improvement
CV (pre-set)	90.1%	70.4%	19.7%
CV (optimized)	93.3%	83.0%	10.3%
Train: *Grid*, Test: *XY*	84.1%	81.8%	2.3%
Train: *Grid*, Test: *XY* *(without room #3)*	97.3%	86.8%	10.5%

TABLE 5.1: Accuracy of the room recognition with and without magnetic field data.

5.2 Room Recognition

The purpose of this evaluation is to check the hypothesis that the magnetic field data improves room recognition accuracy. Additionally, we want to find out which kind of training data (evenly distributed or more samples at the borders) achieves the best results.

To check if the magnetic field data has a significant impact on the performance of the room recognition We perform a 10-fold cross validation[17] on the evenly distributed *Grid* data set both with and without using magnetic field data. The resulting accuracies are compared. This experiment is repeated with different SVM parameters to see if the parameters have an impact on the result.

To determine the best training data set Multiple models are generated using the two training data sets (*Grid*, *Borders*) and different SVM parameters and kernels. The models are tested against the evenly distributed *XY* data set and the resulting accuracies are compared.

For the SVM parameters we use WEKA's standard pre-set values as well as optimized parameters selected with cross validation and grid search.

Results: Magnetic field data Table 5.1 shows a large difference between the performance with and without magnetic field data when using cross validation. With WEKA's pre-set SVM parameters (poly-kernel, $c = 1, e = 1$) the improvement is very large with almost 20%. When using the optimal parameters (determined by grid search for each case separately) the difference drops to a still significant 10%. This drop is due to the fact that the pre-set parameters seem to be better suited to the case with magnetic field data. A comparison between the two cases using not optimal parameters is therefore unfair.

Cross validation can sometimes be a little biased. So the impact of magnetic field data was also compared by training with the *Grid* data set and testing against the *XY* data set, mimicking a real-world use case of the room recognition. Surprisingly this only resulted in a 2.3% improvement.

The reason for this unexpected result can be found by comparing the impact of the magnetic field data on the room recognition accuracy for each room. We see in table 5.2 that the accuracy is improved in every room where it was not already 100%, except for room #3 where the accuracy drops to 0%.

Room Number	with B_{xyz}	without B_{xyz}	Improvement
#1	100.00%	100.00%	-
#2	100.00%	100.00%	-
#3	0.00%	50.00%	-50.00%
#4	100.00%	83.30%	16.70%
#5	100.00%	80.00%	20.00%
#6	100.00%	75.00%	25.00%
#7	100.00%	100.00%	-
#8	83.30%	66.70%	16.60%

TABLE 5.2: Accuracy of the room recognition for each room (*XY* test set, poly-kernel, optimized parameters)

At the time the data sets were collected room #3 was almost empty while the two adjacent rooms contained server racks and technical equipment (near the walls separating the rooms). We expect that some of this equipment caused a disruption of the magnetic field during the time that the *XY* data set was created and therefore obscured the already weak magnetic signature of room #3. A similar effect was observed in related work [12].

When room #3 is excluded from the test set we get the expected 10% improvement when using magnetic field data.

Results: Data sets and SVM parameters Table 5.3 shows the accuracy for the two training data sets and two different kernel functions using WEKA's pre-set parameters. The polynomial kernel performs very well while with the RBF kernel the accuracy is below 20% for all training data sets. The RBF kernel, with these parameters, does not seem to be a good fit for this kind of data.

With the polynomial kernel the unequally distributed *Borders* data performs better. But it is only 2.3% better than *Grid*, which has 40% fewer samples. This difference does not seem significant and is most likely due to the higher numbers of samples.

Table 5.4 shows the accuracy with optimal parameters for each training set and kernel. The parameters were selected with a grid search.

For both training data sets the accuracy could be slightly increased by the parameter selection, but the relative accuracy is still the same; *Borders* has the highest performance with *Grid* only being marginally worse.

With the optimized parameters both kernels have a similar performance, although the RBF kernels c values are generally higher. A high c value means that the RBF kernel's decision surface needs to be more complex and have a smaller margin to achieve the same accuracy as the polynomial kernel. In general a less complex decision surface with a higher margin is preferred. We therefore infer that the polynomial kernel is better suited for this problem.

Finally, to see if it is possible to train a room recognition system with a minimal number of samples the system was trained with the *XY* data set and tested against the *Grid*. This resulted in an accuracy of 81.2%. This is still very high considering only 44 training samples were used.

Training Data (#Samples)	Polynomial $c = 1, e = 1$	RBF $c = 1, g = 0.01$
Grid (223)	81.8%	18.2%
Borders (373)	84.1%	11.4%

TABLE 5.3: Accuracy of the room recognition with different training data sets using the polynomial and RBF kernel presets

Training Data	Polynomial			RBF		
(#Samples)	**accy**	c	e	**accy**	c	g
Grid (223)	84.1%	10	1	84.1%	100	0.01
Borders (373)	88.6%	100	1	86.4%	1000	0.01

TABLE 5.4: Accuracy of the room recognition with different training data sets using the optimal parameters

5.3 Weighting

Section 5.2 has shown that the proposed room recognition can achieve high accuracy even with a small training data set. Potentially it could be trained with the same points used to train the ranging model, eliminating the need to collect additional samples. It is now possible to evaluate the proposed weighting method and see if the information from the room recognition can be used to improve the range based localization accuracy.

In Chapter 3 a new weighting method was proposed; the *Room Weights* (equation 3.1). The goal of this evaluation is to evaluate if the *Room Weights* improve the accuracy of the trilateration and if this is true whether the accuracy can be further improved by combining the *Room Weights* with the existing method of the *Distance Weights*.

In a first step the new weighing method is applied with the assumption of 100% room recognition and compared to *ordinary least squares* (trilateration with no weights). This gives us a best-case value for the improvements with the new weighting method.

In a second step the weighting method is applied to a real word scenario using a room recognition system trained with the *Grid* data set (accuracy of 84%). The results are compared to the 100% case to see how sensitive the weighing method is to errors in the room recognition.

Finally, the performance of the proposed weighting method under real world conditions is compared to the *Distance Weights* to see if it performs better than other simpler weighting methods.

The *ranging model* and *room weights* are trained with the entire *XY* data set and imported into the trilateration tool. For evaluation the same data-set is used.

Results In addition to the mean, standard deviation and maximum error, the results are presented as the cumulative distribution function of the localization error. This allows for a more accurate representation of the performance than just using statistical values.

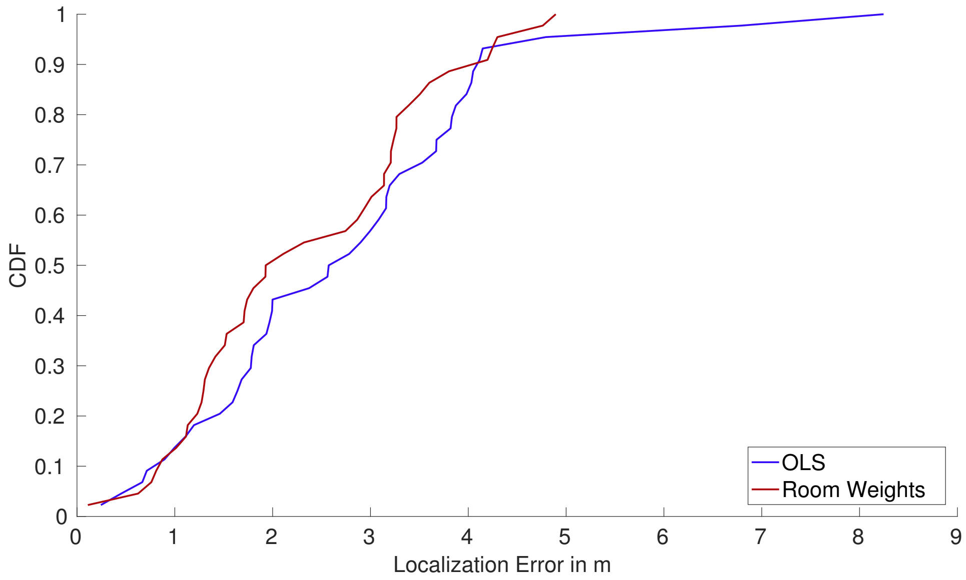

FIGURE 5.3: The Localization error with the *Room Weights*.

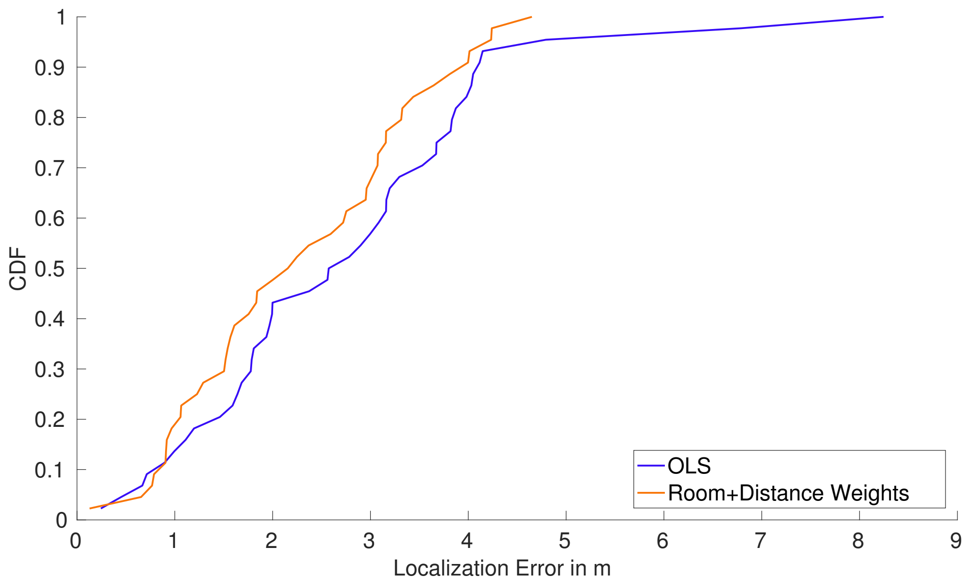

FIGURE 5.4: The Localization error with the *combined room and distance weights*.

Weighting Method	**Mean**	**STD**	**Max Error**	**Improv over OLS**
OLS	2.74m	1.57m	8.25m	
Room Weights (100%)	2.35m	1.22m	4.89m	14.2%
Room+Distance Weights (100%)	2.29m	1.16m	4.65m	16.4%
Room+Distance Weights (84.1%)	2.40m	1.25m	5.69m	12.4%
Distance Weights	2.42m	1.19m	5.37m	11.3%

TABLE 5.5: Comparison of the statistical values for the different weighting methods

The results with 100% room recognition show an improvement of the *Room Weights* over *OLS*. The mean error is 14.2% ($\approx$0.4m) lower, the standard deviation smaller and the maximum error was reduced by $\approx$3.4m. But when looking at the CDF plot (Figure: 5.3) the improvement, although visible, does not seem very dramatic and is mainly due to the large reduction in the maximum error.

The results also show that combining the *Room* and *Distance Weights* does indeed yield a further small improvement. While the mean error is only marginally increased, it lowers the maximum error even further and smooths out the CDF curve (Figure: 5.4).

The comparison between the performance of the weighting in a best-case scenario and real world case, shows that there is indeed an error introduced by the room recognition. But the CDF (Figure: 5.5) shows that this error is overall very small and mainly due to a few samples with a large error, which increase the maximum error.

However, when taking into account this error, the real world performance of the proposed weighting method is almost the same as the existing simpler *Distance Weights*. This is apparent in Figure 5.6. The mean error is only a few centimeters lower (0.07m) and the maximum error even higher than with the *Distance Weights*.

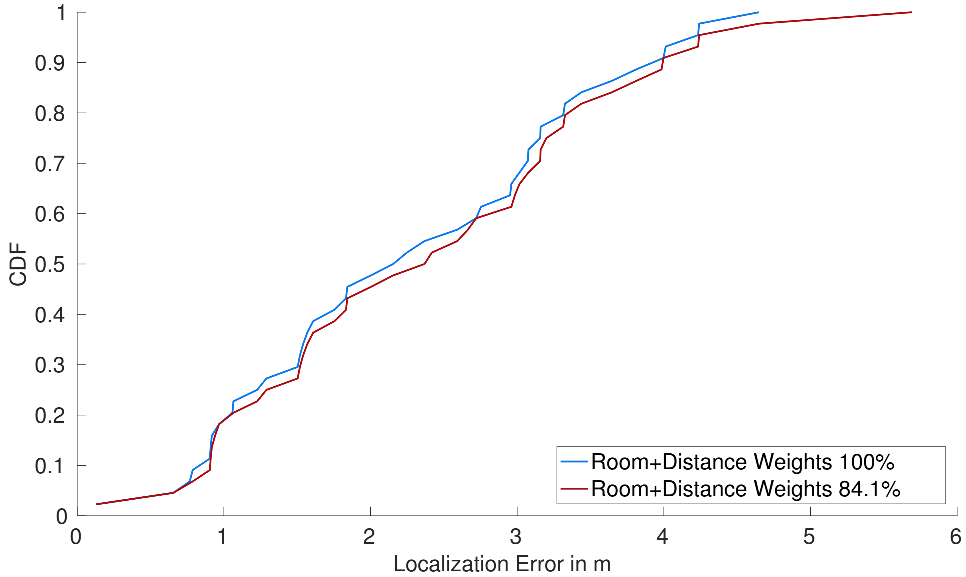

FIGURE 5.5: Localization error with *combined room and distance weights* in a best-case and real world scenario

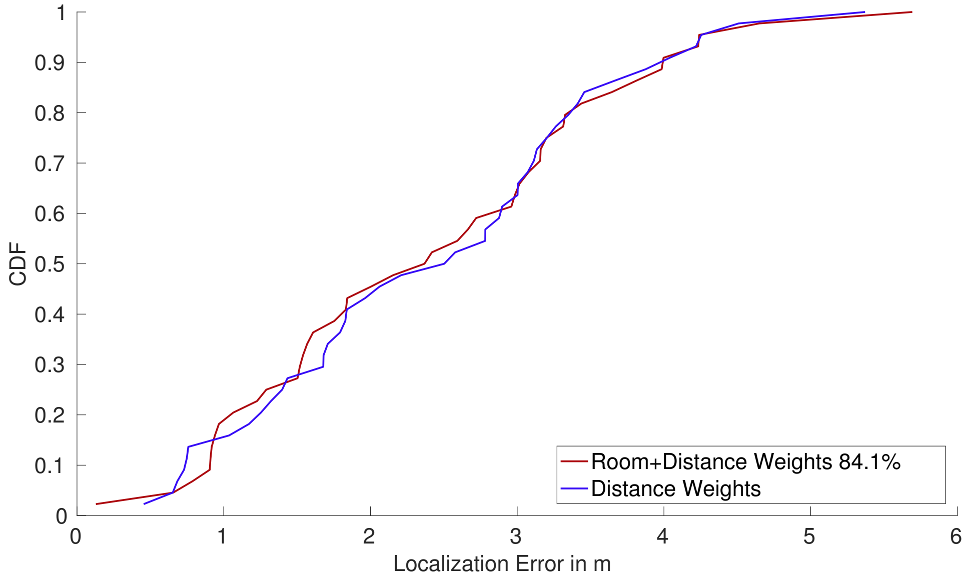

FIGURE 5.6: Localization error for the *combined room and distance weights* and *Distance Weights*

Chapter 6

Conclusion

Indoor localization remains a challenging problem in computer science.

In this thesis we combine the two most common localization approaches into one system i.e., adding a room recognition system to a range based approach to improve the trilateration weights and the localization accuracy.

We propose a room recognition system based on fingerprinting with RSSI and magnetic field data and a new weighting method for range based localization by defining a set of weights for each room.

The proposed system was then implemented in a test bed on a floor of the university building. Several experiments were carried out to evaluate the performance of the room recognition and weighting method.

Concerning the Room Recognition: Room recognition based on RSSI and magnetic field data is able to achieve a high accuracy, even with very small set of training data. The system is able to achieve *80-90% accuracy* depending on the size of the training data set.

The inclusion of magnetic field data *generally improves the room recognition accuracy by 10%*. However, this improvement may be influenced by large temporary disruptions in the magnetic field.

The results of our work indicate that there is no benefit of having a training data set with more samples at the borders. The best results, in comparison to the numbers of samples, are achieved with an *evenly distributed set of samples*.

For the SVM configuration the polynomial kernel seems to be the better suited kernel function.

Concerning the weighting: The proposed weighting method does improve the accuracy compared to *OLS*. However, the improvements are not very large. This can be explained by the fact that the NLR-model used for ranging already takes into account the environmental parameters α and β.

Compared to already existing simpler *Distance Weight* the proposed method performs almost the same. Considering the added complexity and effort in collecting the room recognition samples the proposed weighting method is not practical. It does make more sense to use the *Distance Weights* instead.

Possible future work Although the room recognition was not able to significantly improve the weighting, there are many other possible applications for a simple and effective room recognition system. As an example, it could be included into a particle filter to enhance tracking performance for indoor tracking applications.

Also, to use the room recognition system in real-time, it would be necessary to have a system that takes into account the orientation of the device. The magnetometer readings are dependent on the devices location. So in order to use the device in any orientation, the measurements would need to be normalized to a reference frame. This could potentially be done by keeping track of the devices orientation using the internal sensors and adjusting the magnetometer measurements accordingly.

Bibliography

[1] Muhammad Haris Afzal, Valérie Renaudin, and Gérard Lachapelle. "Assessment of indoor magnetic field anomalies using multiple magnetometers". In: *Proceedings of ION GNSS10* (2010), pp. 1–9.

[2] M. Angermann et al. "Characterization of the indoor magnetic field for applications in Localization and Mapping". In: *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. Nov. 2012, pp. 1–9. DOI: `10.1109/IPIN.2012.6418864`.

[3] Niels Brouwers, Marco Zuniga, and Koen Langendoen. "Incremental wi-fi scanning for energy-efficient localization". In: *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE. 2014, pp. 156–162. URL: `http://ieeexplore.ieee.org/document/6813956/`.

[4] Jose L. Carerra. "Improve trilateration accuracy by LOS/NLOS identification and MIMO". MA thesis. 2015.

[5] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[6] Y. Chapre et al. "Received signal strength indicator and its analysis in a typical WLAN system (short paper)". In: *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. Oct. 2013, pp. 304–307. DOI: `10.1109/LCN.2013.6761255`.

[7] *Cross Validated What is the influence of C in SVMs with linear kernel?* `http://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel`. Accessed: 2016-09-23.

[8] *Eric Kim the Kernel Trick*. `http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`. Accessed: 2016-09-23.

[9] *GitHub leastSquaresTriangulation*. `https://github.com/fischchopf/leastSquaresTriangulation`.

[10] Yanying Gu, Anthony Lo, and Ignas Niemegeers. "A survey of indoor positioning systems for wireless personal networks". In: *IEEE Communications surveys & tutorials* 11.1 (2009), pp. 13–32.

[11] Janne Haverinen and Anssi Kemppainen. "Global indoor self-localization based on the ambient magnetic field". In: *Robotics and Autonomous Systems* 57.10 (2009), pp. 1028–1035.

[12] B. Li et al. "How feasible is the use of magnetic field alone for indoor positioning?" In: *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. Nov. 2012, pp. 1–9. DOI: `10.1109/IPIN.2012.6418880`.

[13] Z. Li et al. "Fine-grained indoor tracking by fusing inertial sensor and physical layer information in WLANs". In: *2016 IEEE International Conference on Communications (ICC)*. May 2016, pp. 1–7. DOI: 10.1109/ICC.2016.7511162. URL: http://ieeexplore.ieee.org/document/7511162/.

[14] Zan Li, Torsten Braun, and Desislava C Dimitrova. "A passive wifi source localization system based on fine-grained power-based trilateration". In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*. IEEE. 2015, pp. 1–9.

[15] Hui Liu et al. "Survey of wireless indoor positioning techniques and systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[16] C Sapumohotti, MY Alias, and SW Tan. "Effects of multipath propagation and measurement noise in IEEE 802.11 g WLAN beacon for indoor localization". In: *PIERS Proceedings*. Vol. 447451. 2012, pp. 27–30.

[17] *Wikipedia Cross-validation*. https://en.wikipedia.org/wiki/Cross-validation_(statistics). Accessed: 2017-05-23.

[18] *Wikipedia Earth's magnetic field*. https://en.wikipedia.org/wiki/Earth's_magnetic_field. Accessed: 2017-03-19.

[19] *Wikipedia Gauss–Newton algorithm*. https://en.wikipedia.org/wiki/Gauss-Newton_algorithm. Accessed: 2017-03-18.

[20] *Wikipedia Levenberg–Marquardt algorithm*. https://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm. Accessed: 2017-05-19.

[21] *Wikipedia Received signal strength indication*. https://en.wikipedia.org/wiki/Received_signal_strength_indication. Accessed: 2017-03-18.