MASTER

Machine Learning for Improved Ultra-wideband Localization

Jin, Xin

Award date:
2020

Department of Electrical Engineering
Electronic Systems Group

Machine Learning for Improved Ultra-wideband Localization

Master Thesis

Xin Jin

Supervisors:
Prof. Guido Dolmans, *Technische Universiteit Eindhoven*
Jac Romme, *Imec, the Netherlands*
Mohieddine El Soussi, *Imec, the Netherlands*

Eindhoven, October 2020

Abstract

The high-accurate indoor localization technique can enable a large number of use cases, such as asset tracking, virtual reality applications, etc.. The ultra-wideband(UWB) transmission is a widely used localization technique and can provide cm-level accuracy in line-of-sight(LOS) environments. However, the UWB signal can be blocked and reflected by the obstacles, and large ranging errors can be introduced in non-line-of-sight(NLOS) environments due to signal attenuation and multipath interference.

In this thesis work, the ranging and localization errors are directly mitigated at the UWB PHY layer in both LOS and NLOS environments. The idea is to extract the representative information from the channel impulse response(CIR) signals and utilize machine learning algorithms to directly estimate the ranging errors to improve accuracy. Several machine learning models such as support vector machine(SVM), k-nearest neighbor(KNN), decision tree and convolutional neural network(CNN), are employed. The proposed models are trained and validated on two data sets. One of the data set was collected in a corridor, while the other was collected in a metallic warehouse environment, so the proposed solution can be validated to be generalized. A LOS/NLOS classifier and an outlier predictor are also introduced in this work, and can provide more prior knowledge for the localization algorithms. Four localization algorithms are going to be introduced and tested. In addition to the traditional methods, an SVM based localization algorithm is proposed and can achieve a higher positioning accuracy.

Preface

It is a great opportunity for me to accomplish the thesis work with many people supporting me.

I sincerely wish to thank Prof. Guido Dolmans for providing the opportunity and resources, and giving me patient guidance during the research process. I also want to thank my mentor Jac Romme for trusting me and providing me with a lot of useful suggestions and kind instructions. Thank you, Mr. Mohieddine El Soussi, for your patient advice and encouragement. By the end, I would like to thank my parents Zanrong Quan and Zhenan Jin, for bringing me to the world and supporting me all the time. Thank you all for your company and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Localization and navigation techniques can provide the position information of moving objects and can enable a myriad of applications. The Global Positioning System(GPS) [1] is a widely used technique for outdoor localization scenarios. The GPS relies on radio signals received from satellites and can provide position information for GPS receivers, which should have an unobstructed line-of-sight to at least four GPS satellites [2]. The traditional GPS can have an accuracy of 5m, and the latest GPS receivers can have a high accuracy of 30cm [3,4]. The GPS performs well in the outdoor environment and can provide critical positioning capabilities to a wide range of applications in both military and commercial sectors. However, the received satellite signals can be obstructed and have a low level of energy in the indoor environment. As a result, GPS is not suitable for indoor localization, and new methods need to be researched.

Indoor localization techniques can provide the ability to locate objects beyond the GPS coverage and enable a large number of applications, such as asset tracking in warehouses, vehicle routing, sports tracking, virtual reality applications, etc. [5–8]. A large variety of wireless technologies can provide solutions for indoor localization. There are several common used localization techniques, such as Wi-Fi, Bluetooth, Zigbee, Radio-Frequency Identification(RFID), Infrared-ray(IR), Ultrasound-Wave, Ultra-wideband etc. [9–12]. Wi-Fi localization technique is based on the measurement of received signal strength(RSS) and the method of fingerprinting, so the line-of-sight is not required. Wi-Fi is cost-effective since it can localize almost every Wi-Fi compatible device without installing extra software, and the coverage range can be 20-50m. However, it can only achieve a ranging accuracy of 2m to 5m [11,13]. The accuracy and power consumption is the main challenge for Wi-Fi localization technique. Bluetooth and Bluetooth Low Energy(BLE) can have characteristics of low pow consumption and low cost. It can provide an accuracy of 2m to 3m within a range of 10-15m [11]. Since a device discovery procedure is needed in each location estimation, the Bluetooth technique has a latency. IR, Ultrasound-wave, and Ultra-wideband techniques can estimate the distance between the fixed base station and the mobile agent based on time of arrival(ToA) or time difference of arrival(TDoA) method and can achieve cm-level accuracy in line-of-sight(LOS) environment. However, the IR signal is easy to be blocked by the obstructions and can not propagate through walls, so it has a short transmission distance of about 5m [14]. The ultrasonic wave can reflect most indoor obstructions, but can attenuate during transmission, which may affect the effective range of the positioning. In this thesis work, we will mainly focus on the ultra-wideband(UWB) indoor positioning technique.

UWB transmission is a widely used technology that can estimate distances between two nodes with cm-level accuracy [15]. UWB can transmit information spread over a large bandwidth in excess of 500MHz [16], and can generate energy in short time duration, so UWB is also known as pulse radio. The short pulses permit a high resolution ranging capability. The UWB signal is also robust and can maintain a high accuracy even in the presence of considerable multipath. UWB can penetrate through the materials such as wood, plastics, glass, brick, etc. [17], and can provide precise localization in complex environments. However, there are several challenges for UWB localization systems. The multi-user interference [18,19] may affect the ranging performance. The UWB signals can be blocked and reflected by obstacles, such as metal, etc.. The none-line-of-sight(NLOS) propagation, as well as multipath effects, can cause a significant deterioration of ranging accuracy. In this thesis study, we will mainly focus on addressing the deficiency and reducing the impact of NLOS propagation by mitigating ranging errors and classify LOS and NLOS signals.

1.2 Related Work

Different position schemes can be applied for UWB localization systems. The localization systems can be implemented based on time of arrival(ToA) [20], while [21] used time difference of arrival(TDoA). Some other works [22] used received signal strength indication(RSS), and [23] used angle of arrival(AoA) to estimate the location. In [24] a combination of ToA and AoA was also implemented, and the hybrid scenario of ToA, TDoA and RSSI was analyzed in [25]. AoA requires at least two base stations, while ToA and TDoA need a minimum of three base stations. ToA and TDoA based systems require synchronization schemes, making the system more complex, but can achieve high accuracy.

Several papers mainly focused on the LOS/NLOS classification. In [26], the NLOS was identified by analyzing the received waveforms. In [27, 28], several features were extracted from the channel impulse response, and the support vector machine(SVM) were utilized for LOS/NLOS classification. In [29], the convolutional neural network(CNN) and long short-term memory(LSTM) were employed for LOS/NLOS signal classification. [30] took a series of CIRs and used recurrent neural network(RNN) to identify the corresponding channel condition. The NLOS signals can be identified and ignored, and mainly use LOS signals for localization. Since only the ranges with small ranging errors are used for localization, the accuracy can be improved. However, it is unsuitable to ignore all the NLOS signals in a complex environment since a large segment of signals are NLOS signals. Several machine learning models can be employed for mitigating ranging error of NLOS signals. In [31–33], several features were extracted manually, and machine learning algorithms such as SVM, Gaussian process and k-nearest neighbors(KNN) were considered for ranging error mitigation. Besides manual feature extraction, several deep learning models, such as CNN and auto-encoder [34,35], were also employed and directly mitigate the ranging error with the CIR signals or I/Q information.

In this thesis work, we extract rich features from the CIR signals and implement a feature selection scheme to select the optimal feature combination. Several machine learning models are proposed to estimate the ranging error with the extracted features, while a CNN model, which doesn't need manual feature extraction, is also proposed. Compared with other works, both simple and complex environments are considered in this work. The proposed solutions are trained and tested on two sets of the database. One was collected in a hospital corridor environment with VNA, while the other was collected in a metallic warehouse with Decawave. Different environments and circumstances are considered and tested in this work to validate the proposed ranging error mitigation solution.

1.3 Research Questions

In this thesis work, we aim at utilizing machine learning algorithms to mitigate the ranging errors and provide a high-accuracy localization solution in the presence of NLOS signals. The research questions are shown as following:

1. What information can we extract from received UWB pulse shape signals to improve indoor localization accuracy?

Many localization algorithms work with distance or angle estimates rather than the received waveform, and therefore valuable information is lost. This thesis work focuses on mitigating ranging errors directly in the physical layer. The interaction of the transmitted UWB pulse shape with the wireless radio channel leads to a distorted received UWB pulse shape. Features from the received signal have to be extracted and utilized for ranging error mitigation and LOS/NLOS classification.

2. Which machine learning algorithm can we use to train the UWB wireless localization link and induce the range and localization errors?

Several machine learning algorithms, such as support vector machine, k-nearest neighbors, decision tree, and CNN are utilized in this thesis work. The models are trained, validated, and tested in different scenarios.

1.4 Thesis Layout

The ranging accuracy performance can deteriorate quickly in the obstructed and reflected environment. This thesis study addresses the problem by utilizing machine learning algorithms to train the UWB wireless localization link. Figure 1.1 illustrates the block diagram of the proposed localization system. The detail of each process will be introduced in the following chapters.

Chapter 2 introduces the preliminaries of UWB localization technology.

Chapter 3 states the representative features that can be extracted from UWB signals and presents four machine learning algorithms that can mitigate ranging errors. Experimental results for ranging error mitigation are also illustrated in this chapter.

Chapter 4 presents the classification algorithm that can be employed for LOS/NLOS classification as well as the outlier prediction. Experimental results for classification models are also shown in this chapter.

Chapter 5 introduces several localization algorithms and illustrates the localization results.

Chapter 6 concludes the results and gives recommendations for future work.

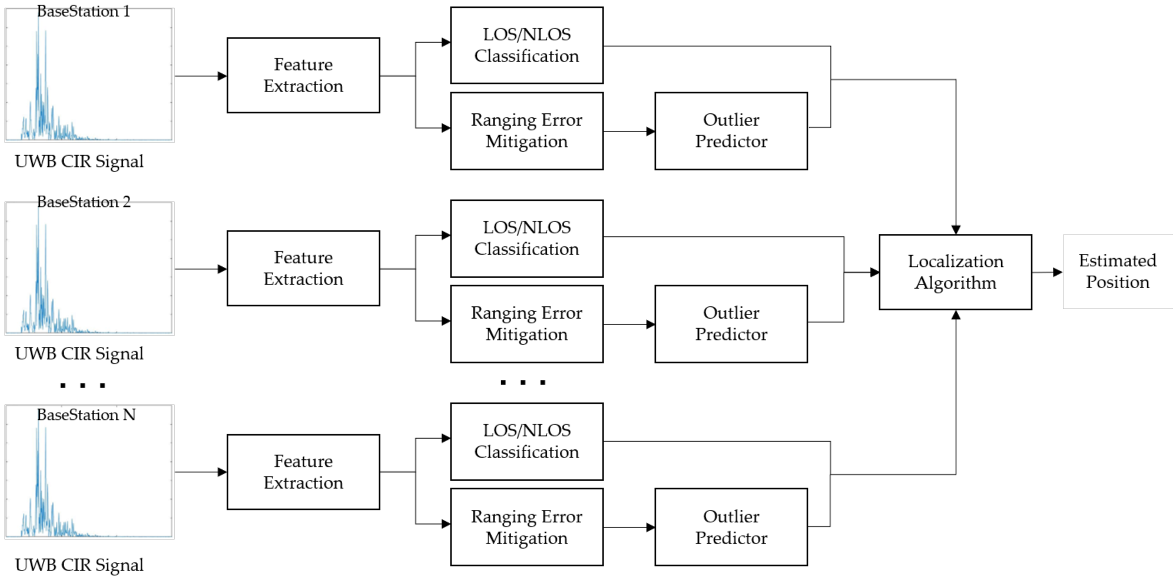


Figure 1.1: Overview of the localization system

Chapter 2

Preliminaries

In this section, we are going to introduce the basic knowledge of the UWB transmission technique. Besides, two sets of the database are trained and tested in this thesis work. The experimental setup will be described in this chapter.

2.1 Introduction to Ultra-wideband

2.1.1 Defination of Ultra-wideband

Ultra-wide Band(UWB) technology is a fast emerging wireless technology and can be widely used in wireless communication, imaging, and positioning systems. The Federal Communication Commission(FCC) in the United States released the first report and order [36] in 2002 and set regulation rules and limits on the emissions of UWB systems. UWB is defined as a transmission system with spectrum occupancy larger than 500MHz and central frequency larger than 2.5GHz, or have a fractional bandwidth more than 20%. The central frequency f_c is defined as (2.1), while the fractional bandwidth B_f is defined as (2.2). FCC also specified a frequency band for the unlicensed use of UWB from 3.1 to 10.6GHz.

$$f_c = \frac{f_H + f_L}{2} \tag{2.1}$$

$$B_f = \frac{f_H - f_L}{(f_H + f_L)/2} \tag{2.2}$$

where f_H and f_L is the upper and lower frequency of the -10 dB emission point.

2.1.2 Characteristics of Ultra-wideband

UWB technology has several features and characteristics that make it a great candidate for positioning systems. The UWB signals have large bandwidth compared with narrow-band signals. Since time and frequency have an inverse relationship, the time duration for UWB signals can be very short and can be capable of resolving multipath components with sub-nanosecond delays. As a result, UWB signals can present a high time resolution and have a low susceptibility to multipath interference and fading.

UWB can also obtain a high data rate and capacity. The Shannon-Hartley theorem [37] defines the upper bound of the data rate that can be achieved relationship and indicates a direct relationship between the capacity and bandwidth. The Shannon-Hartley theorem is given by:

$$C = B\log_2(1 + \frac{S}{N}) \tag{2.3}$$

where C is the channel capacity, B is the channel bandwidth, S is the average power of the received signal over the bandwidth, N is the average power of the noise over the bandwidth. S/N is also denoted as the signal to noise ratio(SNR). According to the theorem, we can find that a low power consumption can be achieved for a specific capacity. Besides, due to the large bandwidth of UWB signals, the system can still obtain a high information data rate from 100 to 500Mbps [16] in the low power spectral density circumstances.

UWB signals can overlay and coexist with the already available communication services in the specified frequency band. FCC regulates the upper bound of the power level of -41.3 dBm/MHz. The narrowband signals in the same frequency range, such as IEEE 802.11 wireless local area networks, may present a higher power spectral density compared with that of UWB. As a result, UWB can coexist with other wireless communication networks in the UWB frequency range.

2.1.3 Ranging Estimation

In this thesis work, we utilize the time of arrival(ToA) method to estimate the distance between the mobile agent to the fixed base station. ToA is defined as the absolute time instance from the signal emanated by the transmitter to the signal received by the receiver, which also can be called as time of flight(ToF). The distance can be directly calculated from ToA with the speed of the light.

2.2 Experiment Environment

In this paper, two UWB datasets are used to train, test and validate the proposed solutions. The first database was measured in a hospital corridor, and is utilized for training and testing the whole positioning system, including ranging error mitigation, LOS/NLOS classification, outlier detection, and localization estimation. The second database was measured in a warehouse with metal obstacles and is mainly used in ranging error mitigation part, and validate the proposed ranging error mitigation solutions.

2.2.1 Database 1

The first database was measured and collected in a hospital environment. The measurements were conducted on the second floor of the B wing of the Kennedy campus of the AZG, Kortrijk

Belgium [38]. The environment floor plan is shown in Figure 2.1. There was a straight corridor connecting six bathrooms, three double-rooms, three hallways, and a small utility room. All doors remained open during the measurement procedure. Nine anchor antennas were implemented, and were located at the yellow points as shown in Figure 2.1. A mobile tag was placed at 92 cm above the floor, and moved along the red route as shown in the map with a spacing of 10 cm. The route and consists of 21 segments and can cover a distance of over 40m.

The signals were measured with a 4-port VNA from the frequency domain. The signals were measured with a carrier frequency of 7.5GHz and bandwidth of 5GHz at 4096 discrete frequency points. The signal from the frequency domain should multiply a Hann window. The Hann window can reduce the -3dB bandwidth, resulting in the largest system bandwidth of 1.82 GHz.The trimmed frequency domain measurements were transferred to the time domain with inverse Fourier transform, and channel impulse response(CIR) signals could be obtained. The CIR signals can be used for feature extraction and model training and validation. Each anchor implemented a threshold detector, and could record the delay index of the first signal path that crosses the threshold with respect to the transmit time of the agent, which can also be called as ToA. Then the ranges could be estimated and reported based on ToA.

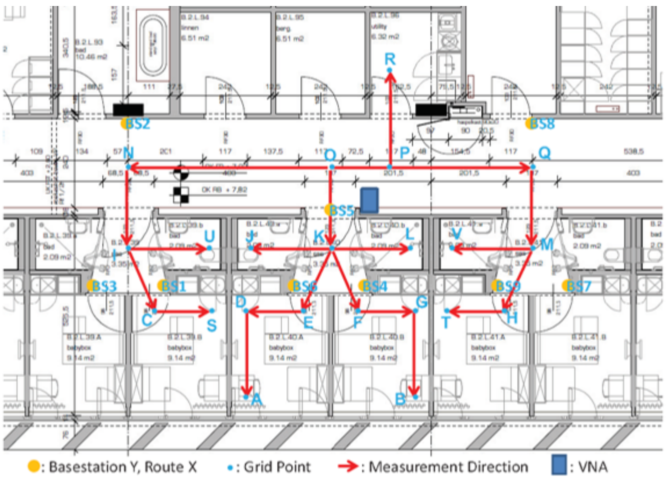
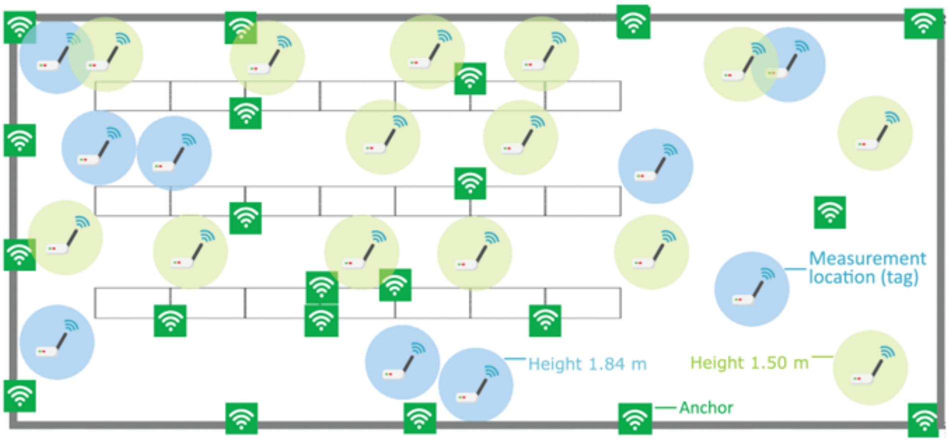


Figure 2.1: Environment floor plan of database 1

2.2.2 Database 2

In this work, a second database is used to validate the proposed ranging error mitigation algorithms. The second database was collected in IDLabs [39]. The experimental setup details can also be found in [34]. Figure 2.2 shows the floor plan and pictures of the experimental environment. There were three metal racks in the middle of the testbed, which can obstruct and reflect the UWB signals. In total, 21 base stations were implemented with almost the same height, and they could cover multiple positions in the experimental environment. 23 locations scattered in the different positions and height(around 1.5m and 1.84m) were tested to introduce the variety in the database. In total, 28473 ranging measurements were collected, among which 18796 samples were LOS signals and 9677 samples were NLOS signals.

The device used for data collection [40] was implemented based on UWB Decawave 1000 transceiver. It can provide CIR signals with a sequence of 1016 complex numbers, and the CIR index is approximately 1ns long. The distance could be estimated with ToA and reported by the DW 1000 transceiver.



(a) Floor plan of the measurement environment. The green wireless symbol denotes the location of base stations, while the light-green circles are the measurement locations with a height of 1.50m, and the blue circles are the measurement locations with a height of 1.84m.

(b) Picture of the metallic warehouse

Figure 2.2: Floor plan and picture of experimental environment for database 2

Chapter 3

Ranging Error Mitigation

Ultra-wideband technology can achieve cm-level accuracy under line-of-sight circumstances. However, in the none-line-of-sight situation, ranging accuracy can degrade significantly due to obstruction and reflection. In this chapter, we are going to extract features from UWB signals and utilize machine learning algorithms to mitigate ranging errors. Four machine learning models are going to be introduced, which are support vector machine(SVM), k-nearest neighbors(KNN), decision tree and convolutional neural network(CNN). These models will be trained, tested and compared on two datasets as described in Section 2.2.

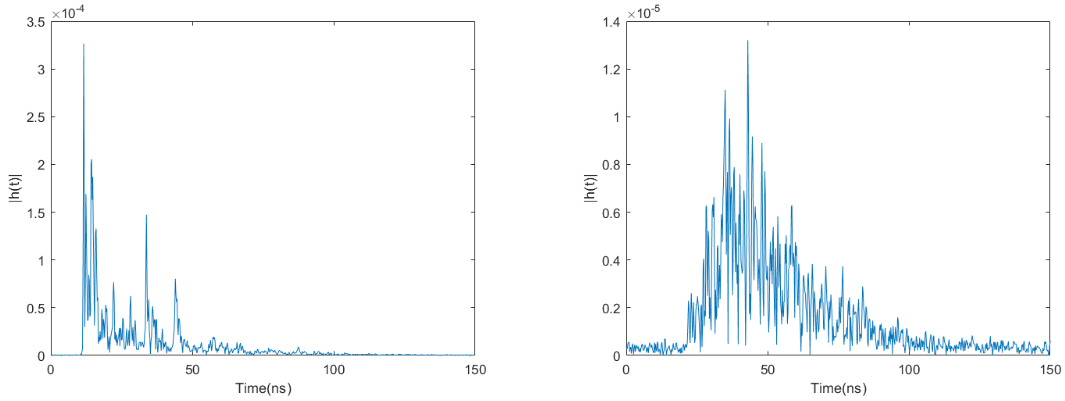
3.1 Feature Extraction

If no obstructions exist in the straight wireless line between the transmitter and the receiver, it can be considered as line-of-sight(LOS) case. Otherwise, it should be regarded as none-line-of-sight(NLOS) case. The range between two nodes can be measured based on the ToA method. A simple threshold-based method can be employed to detect the arrival time of the first path signal.

Figure 3.1 shows example channel impulse response(CIR) signals in LOS and NLOS conditions, respectively. In most LOS cases, the first arrival path is the strongest path. Thus it can achieve a high ranging accuracy with the ToA-based method. In the NLOS cases, UWB signals are attenuated due to obstruction, and multipath components may arrive before the strongest path due to reflection, which can result in large ranging error.

The differences between the LOS signals and NLOS signals are intuitively obvious. Machine learning algorithms can be implemented for LOS/NLOS classification and ranging error mitigation. We need to compute statistic parameters from CIR signals as input features for the machine learning algorithm. In reference [31], seven representative features are selected to represent the UWB signals, which are energy, maximum amplitude, root mean square delay spread, mean excess delay, rise time, kurtosis and estimated distance. In this paper, three additional features, which are standard deviation, signal-to-noise-ratio, rician K factor, are extracted to describe the signal from various perspectives and improve the performance.

The first two features are energy and maximum amplitude. In NLOS cases, the UWB signals



(a) LOS

(b) NLOS

Figure 3.1: The absolute channel impulse response(CIR) of LOS/NLOS signals

can be attenuated due to obstruction. As a result, energy and maximum amplitude of the NLOS signal are smaller compared with that of LOS signals.

Energy:

$$E = \int_T r(t)^2 \, \mathrm{d}t \tag{3.1}$$

where $r(t)$ is the received signal.

Maximum amplitude:

$$r_{max} = max_t |r(t)| \tag{3.2}$$

Standard deviation is a statistic parameter to describe the shape of received CIR signal, which is given by (3.3).

Standard deviation:

$$\sigma = \sqrt{\frac{1}{T} \int_T (|r(t)| - \mu)^2 \mathrm{d}t} \tag{3.3}$$

where μ is the mean value of the received signal, and $\mu = \frac{1}{T} \int_T |r(t)| \mathrm{d}t$.

Kurtosis can describe the steepness of received signals. The received signals with high kurtosis tend to present a significant direct path component and weak multipath components, and exhibit a concentrated energy distribution. The received signals with low kurtosis tend to present a flat energy distribution, which means the direct path component is less significant than multipath components.

Kurtosis:

$$\kappa = \frac{1}{\sigma^4 T} \int_T (|r(t)| - \mu)^4 \, \mathrm{d}t \tag{3.4}$$

The Rician K factor is defined as the ratio between the deterministic signal power and the variance of the multipath [41]. K factor can specify the Rician distribution, which is given by (3.5).

Rician K factor:

$$K = \frac{r_{max}^2}{2\sigma^2} \tag{3.5}$$

The signal-to-noise-ratio(SNR) can be used to compare the level of meaningful signal to the level of background noise. Since the NLOS signals are scattered due to reflection and obstruction, the SNR of NLOS signals can be lower than LOS signals.

Signal to noise ratio:

$$SNR = \frac{P_s - P_n}{P_n} \tag{3.6}$$

where P_s denotes the power of received signal, and $P_s = \frac{1}{T}\int_T r(t)^2 \mathrm{d}t$. P_n denotes the power of background noise. Since each received UWB signal is followed by a noise-only part, P_n can be calculated with the noise-only signal, and $P_n = \frac{1}{T}\int_{T_N} n(t)^2 \mathrm{d}t$, where $n(t)$ is the noise-only part signal.

Features mentioned above can provide information about the amplitude statistics of the received CIR signal. The delay properties of the received CIR should also be considered. Mean excess delay(MED) and root mean square(RMS) delay spread can characterize the temporal desperation information caused by multipath components. Compared with the LOS signal, the NLOS signal can have more multipath components due to the reflective environment, and represent larger MED and RMS values.

Mean excess delay:

$$\tau_m = \frac{\int_T (t \cdot |r(t)|^2)\, \mathrm{d}t}{E} \tag{3.7}$$

RMS delay spread:

$$\tau_{RMS} = \frac{\int_T (t - \tau_m)^2 \cdot |r(t)|^2\, \mathrm{d}t}{E} \tag{3.8}$$

Rise time is defined as the time duration between the arrival time of the first path and arrival time of the strongest path. For the LOS signal, the first path is the strongest path and the signal can present a short rise time. For NLOS signals, multipath components can arrive earlier than the strongest path. Thus the rise time is longer for NLOS signals.

Rise time:

$$t_{rise} = t_H - t_L \tag{3.9}$$

where t_L denotes the arrival time of the first signal that can pass the noise level, and $t_L = min\{t : |r(t)| \geq \alpha\sigma_n\}$. σ_n is the standard deviation of background noise and α should be set to a value larger than zero to avoid false alarm. t_H denotes the arrival time of the strongest path, which is defined as $t_H = min\{t : |r(t)| \geq \beta r_{max}\}$, and $0 < \beta < 1$. The values of α and β should be set to the suitable value according to the used database in practice. After testing on the data set, we set α as 3 and β as 0.9 in our case.

The last feature is the distance estimated with the threshold-based ToA method. The estimated distance may contain a large ranging error, especially for NLOS cases, but can provide reference information for the ToA of the direct path and should be an important feature.

Based on the feature extraction, a feature vector $\mathbf{X}$ can be obtained by each received CIR signal and can be used as the input vector for regression and classification algorithms. The feature vector is denoted as:

$$\mathbf{X} = [E, r_{max}, \sigma, \kappa, K, SNR, \tau_m, \tau_{RMS}, t_{rise}, \hat{d}] \tag{3.10}$$

3.2 UWB Ranging Error Estimation Algorithms

A regression algorithm can create a model that can predict the continuous quantity output with the input features. As described in Section 3.1, we can extract ten features from the CIR signals and use these features as input variables of a machine learning regression algorithm to estimate ranging errors. In this section, four classic machine learning regression algorithms, which are support vector machine(SVM), k-nearest-neighbors(KNN), decision tree and convolutional neural network(CNN) are going to be introduced.

3.2.1 Support Vector Machine Regression

Support vector machine is a supervised machine learning algorithm [42, 43]. Suppose that in a regression problem, the input data set is $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in R^n$ and $y_i \in R$, and N is the number of input samples. The goal is to learn the mapping function from input variable $\mathbf{x}_i$ to output variable y_i. Firstly, a SVM regressor can make an optional feature transformation, and transform the input variable x into $\varphi(\mathbf{x})$. Assumed that it is a linear regression problem, and the relationship between output variable and input variable is given by

$$y(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \tag{3.11}$$

where $\varphi(x)$ is the optional transformation of the input feature, and w is the weight matrix and b is the bias. The unknown variable w and b should be determined by training the data set.

The mapping function of the SVM regressor can be interpreted as a hyperplane as defined in 3.11. Assume that there exists two bounding hyperplanes $y(\mathbf{x}) + \epsilon = 0$ and $y(x) - \epsilon = 0$ ($\epsilon > 0$). The Euclidean distance between two bounding hyperplanes is $d = 2\epsilon/\sqrt{||\mathbf{w}||^2 + 1}$. The idea of the SVM regressor is to maximize the margin between two bounding hyperplanes so that the ϵ-tube can fit as many data points as possible, hence the formulation of this optimization problem can be stated as following:

$$\underset{\mathbf{w},b}{\arg\min} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} E(y(\mathbf{x}_i) - y_i) \tag{3.12}$$

where $E(y(\mathbf{x}) - t)$ is the penalize error function, and C is the penalty parameter that can control the trade-off between maxmizing the margin and minimizing training error.

In support vector machine, an ϵ-insensitive error function is employed. If the absolute difference between the prediction $y(\mathbf{x})$ and the target t is less than a setting value ϵ where $\epsilon > 0$, the error can be considered as zero. The ϵ-insensitive error function also implies that only the data points outside the ϵ-tube can contribute to the error function, so it can be called support vectors. An example of ϵ-insensitive error function with the linear cost is given by

$$E_\epsilon(y(\mathbf{x}_i) - y_i) = \begin{cases} 0, & if|y(\mathbf{x}_i) - y_i| < \epsilon \\ |y(\mathbf{x}_i) - y_i| - \epsilon, & otherwise \end{cases} \tag{3.13}$$

The optimization problem can be transformed into a dual problem, and the regression expression can be solved as following:

$$y(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i) + b \tag{3.14}$$

where $\kappa(x, x_i)$ is the kernel function. In most cases, the ideal hyperplane is not a simple linear hyperplane. The input data should be transformed into a high-dimensional space, and a complex non-linear hyperplane can be utilized to fit the data points. The use of the kernel trick can eliminate the need for direct data transformation with $\varphi(\mathbf{x})$ function. The kernel trick can return the scalar product of $\varphi(\mathbf{x}) \cdot \varphi(\mathbf{x}')$ directly, which allows us to transform the data into high-dimensional spaces without the penalty of excessive computational costs.

There are three commonly used kernels [42]: linear kernel, polynomial kernel and radial basis function(RBF) kernel.

The linear kernel simply returns the scalar product between two points. This results in a linear hyperplane. The linear kernel function is given in (3.15). The linear kernel has the lowest cost and it is faster to train a support vector machine with a linear kernel compared with other kernels.

$$\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}' + 1 \tag{3.15}$$

The polynomial kernel allows for curved hyperplanes. The kernel function is given by:

$$\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^d \tag{3.16}$$

where the exponent d indicates the degree of the polynomials.

The RBF kernel specifies the similarity between the two data points, which is given by:

$$\kappa(\mathbf{x}, \mathbf{x}') = exp(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}) = exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2) \tag{3.17}$$

where $||\mathbf{x} - \mathbf{x}'||^2$ is the squared Euclidean distance between two data points, and the parameter γ is used to control the smoothness of the boundary. If the value of γ is too small, the hyperplane will be smooth and may cause underfitting. If the value of γ is too large, it can cause an overfitting problem. Thus, it is important to set a suitable value for γ.

3.2.2 K-Nearest Neighbors for Regression

SVM regression can estimate and mitigate the ranging errors effectively, but it requires a complex computation in the process of finding the hyperplane that can maximize the distance between two bounding hyperplanes. K-nearest neighbors(KNN) is one of the simplest non-parametric learning method and widely used in classification and regression problems [44]. The idea of KNN is to calculate the distances between the data point and all the observation examples in the data set, choose Ks nearest examples, and gather the information to estimate the output value. Compared with SVM, KNN is more suitable for low complexity and less intensive computational requirements.

Suppose that the training database has N samples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i$ is the feature vector and y_i is the corresponding expected output. The Euclidean distance between each observation examples $\mathbf{x}_i$ and data point $\mathbf{x}$ is given by

$$D_i = ||\mathbf{x} - \mathbf{x}_i|| = ((\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i))^{1/2} \tag{3.18}$$

The observations can be ranked according to the distances, which is $\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, ..., \mathbf{x}_{(N)}\}$, while $\mathbf{x}_{(k)}$ is the kth nearest neighbor of the data point $\mathbf{x}$.

The classical KNN estimate of the output $\mathbf{y}$ is the average value of the output among the k nearest observations of the data point $\mathbf{x}$, which can be given by:

$$y = \frac{1}{k}\sum_{i=1}^{k} y_{(i)} \tag{3.19}$$

A widely used extension of the classical KNN estimate is the weighted KNN regression. This allows closer neighbors can have a greater influence on the result compared with neighbors, which are further away. The weighted KNN estimate is defined as:

$$y = \frac{\sum_{i=1}^{k} \frac{1}{D_{(i)}} y_{(i)}}{\sum_{i=1}^{k} \frac{1}{D_{(i)}}} \tag{3.20}$$

3.2.3 Decision Tree for Regression

The Decision tree is a commonly used practical approach for supervised learning [45]. The idea of the decision tree is to build a tree structure and break down a data set into smaller subsets that contain instances with similar values. A decision tree consists of three types of nodes. The root node is the topmost decision node, which represents the entire sample and may get split further. The interior node represents features of a data set and can have branches that represent the decision rules. The leaf node represents a decision on the numerical target.

With a particular data point, it runs completely through the entire tree by following the decision rules until it reaches the leaf node. The final prediction is the average value of the data in the particular leaf node. An example of the decision tree model is shown in Figure. 3.2.

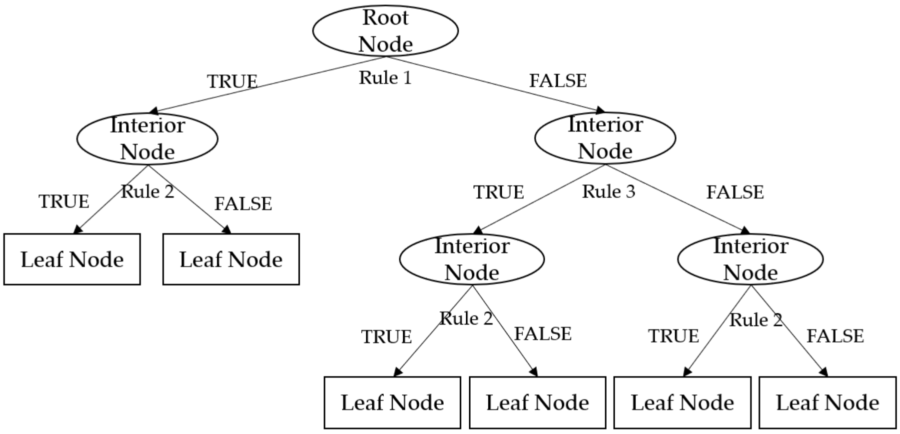


Figure 3.2: Decision Tree Model

3.2.4 Convolutional Neural Network for Regression

In the previous sections, we manually extract ten representative features from the CIR signals and apply machine learning algorithms to learn the mapping function from the input features to ranging errors. The CIR signal can also contain latent representative features that have not been explored. In this section, we are going to introduce the convolutional neural network(CNN) to learn from the latent features and estimate ranging errors.

The architecture of CNN [46] is an extension of the multilayer perception(MLP) model, which includes a number of convolutional layers typically as hidden layers. Figure 3.3 demonstrates the architecture of an example CNN model. The convolutional layers apply a convolution operation to the input. These layers can detect latent feature patterns in the input images or arrays. The convolutional layers can be optionally followed with pooling layers. The pooling layers can sub-sample the values of a local region into a single value and reduce the size of the output. The maximum pooling is the most commonly used pooling layer. This layer selects a specified window size and produces the maximum value in the window as the output for that location. The sequence of convolutional layers combines these patterns in increasingly complex patterns or higher-level features. Finally, the patterns need combined and mapped to the target values. In order to achieve this, we need to flatten the output of the last convolutional layer and feed that to a dense layer, which is a fully connected layer. Both convolutional layers and dense layers have an activation function.

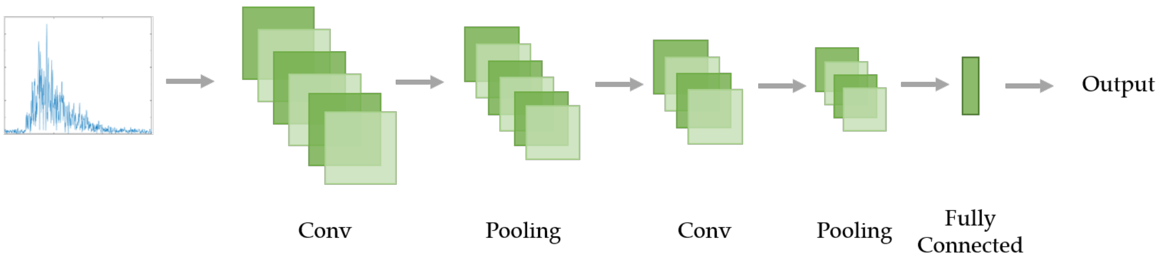


Figure 3.3: Architecture of CNN model

3.3 Procedure of Ranging Error Estimation

1. Data Preparation

For each CIR signal sample, ten features can be extracted as described in the previous section. The true ranging error, which is defined as $\Delta = \hat{d} - d$, should also be recorded for supervised learning, where $\hat{d}$ is the distance estimated with ToA, and d is the true distance. The goal is to learn the mapping function from the input features to the ranging errors. Since the input features are in various scales, the features with larger values can greater influence on the model compared features with small values. As a result, it is essential to normalize the input features and transform the values into a common scale.
The inputs for the CNN model are CIR samples. The noise-only parts are removed, and only the signal parts which contain useful information are trimmed. The trimmed signals also need to be normalized and act as the input of the CNN model.

2. Model Training

The database is divided into the training set and the test set. The training set can be used to fit the parameters of the model. When we fit the model with the training set, the hyperparameters of the model need to be tuned, and the validation set can provide an unbiased prediction of the performance of the tuned model on another dataset. In this project, a 5-fold cross-validation method is employed. The training set can be evenly divided into five disjoint subsets. For each iteration, four subsets act as the training set, while the remaining subset acts as the validation set. The average performance of the validation sets will be considered as the performance of the whole training set. The performance of the validation set is an important criterion when fine-tuning the hyperparameters.

3. Model Evaluation

After fitting the mapping function with the training set, we can use the test set to evaluate the proposed model. The performance is evaluated based on the cumulative distribution function(CDF), mean error, and mean absolute error(MAE) of the residual ranging error. The residual ranging error is defined as $\Delta_i - \hat{\Delta}_i$, where $\hat{\Delta}_i$ is the estimated ranging error of the ith sample, and Δ_i is the actual ranging error. The CDF can calculate the cumulative probability for the residual ranging error. The mean absolute error can describe the ranging accuracy, and is defined as

$$MAE = \frac{\sum_{i=1}^{N} |\Delta_i - \hat{\Delta}_i|}{N} \tag{3.21}$$

3.4 Ranging Error Estimation Result

3.4.1 Ranging Error Estimation Result on Database 1

1. Ranging Error Estimation Result

In this project, 70% of the samples are random selected as the training set, while the remaining 30% samples are the test set. It is guaranteed that the proportion of LOS and NLOS signals is the same in the training set and test set. By utilizing the cross-validation method, the hyperparameters for SVM are set to the optimal values. The SVM regression model for ranging error estimation uses RBF kernel function, and the kernel coefficient parameter γ is set to 0.1. The penalty parameter C is set to 10, while the parameter ϵ for ϵ-insensitive error function is set to 0.1.

Figure 3.4 shows the CDF of residual ranging error after mitigating the ranging error with SVM, while the carrier frequency is 7.5GHz and the bandwidth is 1.82GHz. The MAE for original unmitigated data is 1.90m, and 90% of the ranges errors can be smaller than 4.61m. The ranging distance is estimated by setting a threshold to detect the ToA of the first path. The threshold is set to a suitable value to avoid early detection, for instance detect a noise spike as the first path. As a result, most of the unmitigated ranging errors are positive values. Since most of the errors are positive value, the average performance can be improved with an optimal bias correction. The 50% of the CDF for unmitigated ranging errors is 1.33m, and can be considered as a bias. If we subtract the bias of 1.33m from all ToA estimated ranges, an optimal mean error of 0.56m and an MAE of 1.47m can be achieved.

By mitigating the ranging errors with the SVM regression model, the mean error is 0.04m, while the MAE can be reduced to 0.32m, and 90% of the ranging can be reduced to an absolute error smaller than 0.87m. By discovering the ranging error mitigation performance of LOS and NLOS signal respectively, we can find that the MAE of NLOS is improved significantly from 2.61m to 0.46m, while the MAE of LOS is reduced from 0.32m to 0.12m. Compared with the residual ranging error improved with optimal bias correction, the SVM can indeed improve the ranging accuracy.

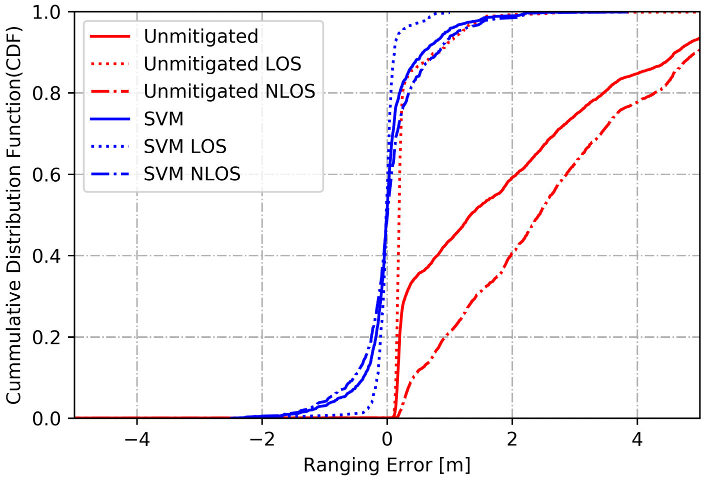


Figure 3.4: CDF of unmitigated ranging error and ranging error mitigated with SVM. The carrier frequency is 7.5GHz, and the bandwidth is 1.82GHz.

Figure.3.5 compares the ranging error estimation performance of the test set among SVM, KNN, decision tree and CNN. For the KNN model, the parameter k, which indicates the number of neighbors used for k neighbors queries, is set as 3. The depth of the decision tree is set as 10, and the minimum number of samples required to split an internal node is 20. The model summary for CNN can be found in Table 3.1. The SVM can obtain a mean error of 0.04m and MAE of 0.32m, while the performance of KNN is slightly deteriorated, which has a mean error of 0.03m and MAE of 0.33m. The mean error and MAE for decision tree can reach 0.04m and 0.35m. CNN can reach a mean error of 0.20m and MAE of 0.47m. All these models can improve the ranging accuracy significantly. SVM can achieve the best performance among these models and should be the optimal option for ranging error estimation. KNN can reach a similar result as that of SVM, and have a shorter training time and lower computation complexity. As a result, KNN can also be a suitable option for low complexity and computational capabilities networks.

Since the samples in the training set and test set are selected randomly, the data points in each set can spread in the whole map, and the position of the data points in the test set can be close to that in the training set. It is also essential to separate the training set and test set based on positions. We can select the test set as shown in Figure 3.6. The remaining part is considered as the training set. In the test set 1, nine routes are considered as unknown positions, and the remaining routes can be used for training the model. Three routes are selected as unknown positions in the test set 2, and the remaining part can be the training set.

Table 3.1: Architecture of CNN model

Layer	Output dimension
Input	1x500x2
Conv(1,7)	1x500x32
Maxpooling(1,3)	1x167x32
Conv(1,3)	1x167x96
Maxpooling(1,3)	1x56x96
Conv(1,3)	1x56x64
Maxpooling(1,3)	1x19x64
Fully Connect Layer	
Fully Connect Layer	
Fully Connect Layer	
Output	1

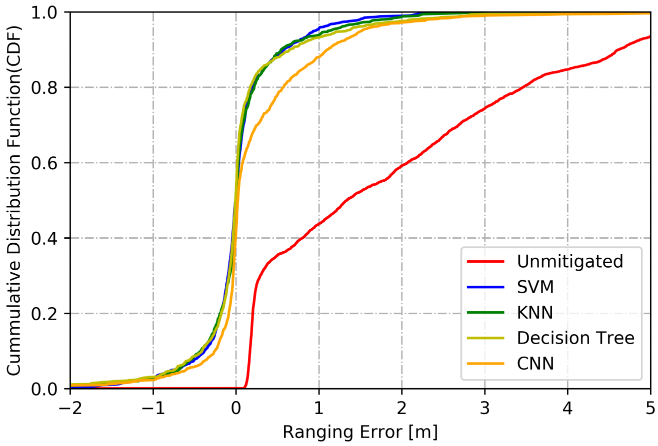


Figure 3.5: CDF of unmitigated ranging error and ranging error mitigated with SVM, KNN, Decision Tree and CNN. The carrier frequency is 7.5GHz, and the bandwidth is 1.82GHz.

The ranging error estimation performance on these two test sets is shown in Figures 3.7 and 3.8. By selecting the test set 2, SVM can achieve a mean error of 0.11m and MAE of 0.33m, while the MAE of KNN and decision tree are degraded to 0.40m and 0.46m, respectively, and the mean error for KNN and decision tree is 0.02m and -0.04m. The mean error for CNN is -0.17m, and the MAE is 0.46m, which is similar to the previous result. By selecting the test set 1, more positions are considered as test set and the number of samples in the training set is decreased. The SVM and CNN can still maintain the mean error of 0.05m and 0.07m, and the MAE of 0.31m and 0.40m, respectively. The performance of KNN and decision tree are degraded. The KNN has a mean error of -0.05m and MAE of 0.44m, while the decision tree has a mean error of -0.15m and MAE of 0.52m. If the training data is sufficient enough to cover the whole map in the experimental environment, all these models can achieve a similar and high ranging accuracy. However, when we set several parts as known positions, SVM and

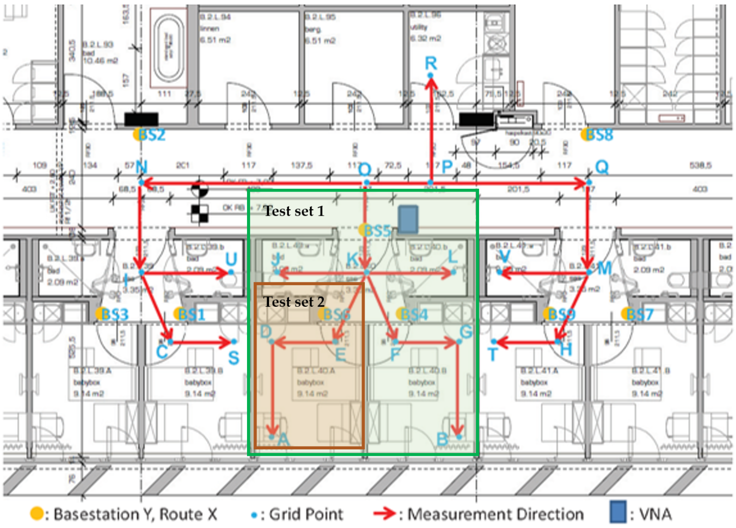


Figure 3.6: Test Set Selection

CNN can still keep a stable ranging error mitigation performance, while the performance of KNN and decision tree are degraded significantly. As a result, SVM is a suitable option when the training data is not sufficient enough. Since the SVM can have the optimal and stable performance in both circumstances when the test set is scattered randomly in the experimental environment or the test set is in unknown routes, SVM can be considered the optimal option for ranging error mitigation for this data set and will be used for further analysis.

2. Feature Selection

As stated in section 3.1, ten features are extracted from the CIR signals and act as the input of machine learning algorithms. From the test results, we can find that the SVM can present the best performance among the selected machine learning algorithms. In this section, we are going to test all the feature combinations on SVM to explore the most suitable feature combination for ranging error estimation.

In this experiment, 30% of the data sets are randomly selected as the test set, while the remaining 70% are the training set. The performance is evaluated based on MAE. The performance of the top 10 feature combinations is shown in Table 3.2. The features are denoted as numbers in the table, and the number 1-10 are corresponding to energy, maximum amplitude, standard deviation, kurtosis, rician K factor, signal to noise ratio, mean excess delay, RMS delay spread, rise time, and estimated distance, respectively.

From the table, we can find that the ranging error mitigation performance is better when we introduce more features since the CIR signals can be described more precisely with more features. The combination of all the ten features is proved to result in the best performance.

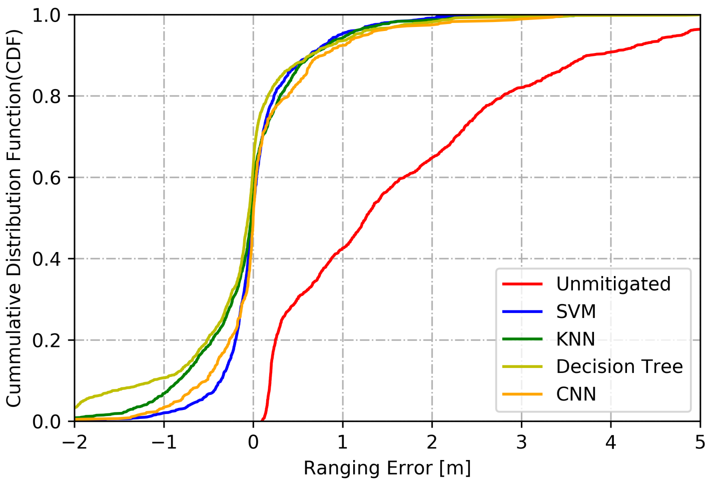


Figure 3.7: Ranging error mitigation performance of SVM, KNN, CNN and decision tree on test set 1. The carrier frequency is 7.5GHz, and the bandwidth is 1.82GHz.

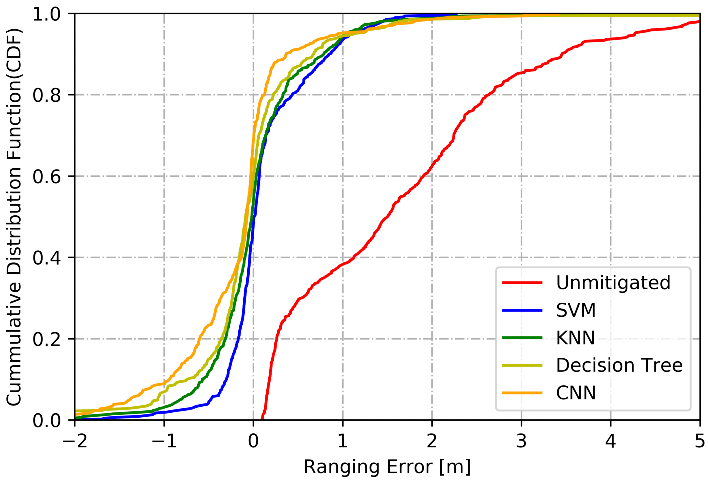


Figure 3.8: Ranging error mitigation performance of SVM, KNN, CNN and decision tree on test set 2. The carrier frequency is 7.5GHz, and the bandwidth is 1.82GHz.

Table 3.2: Ranging Error Estimation Performance for Different Feature Combinations

Feature Combination	Number of Features	MAE (m)
1,2,3,4,5,6,7,8,9,10	10	0.3179
1,3,4,5,6,7,8,9,10	9	0.3181
2,3,4,5,6,7,8,9,10	9	0.3189
3,4,5,6,7,8,9,10	8	0.3194
1,2,4,5,6,7,8,9,10	9	0.3164
1,4,5,6,7,8,9,10	8	0.3206
2,4,5,6,7,8,9,10	8	0.3214
1,2,3,4,5,7,8,9,10	9	0.3216
4,5,6,7,8,9,10	7	0.3219
1,3,4,5,7,8,9,10	8	0.3221

3. Ranging Error Estimation Performance for Signal with Different Bandwidth

UWB signal with large bandwidth leads to a high time resolution of CIR signal and can contain more information and distinguish the leading path from multipath components. Figure 3.9 shows ranging error mitigation performance of UWB signals with different bandwidth. By decreasing the bandwidth from 1.82GHz to 500MHz, the MAE is increased from 0.32m to 0.36m. The ranging accuracy can be slightly influenced by changing the bandwidth.

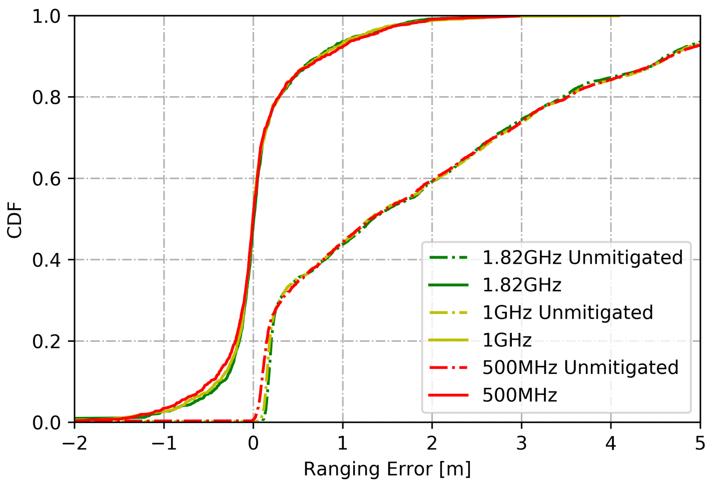


Figure 3.9: CDF of unmitigated ranging error and ranging error mitigated with SVM. The carrier frequency is 7.5GHz.

3.4.2 Ranging Error Estimation Result on Database 2

In this section, we are going to validate the proposed ranging error mitigation algorithms on the second data set, which was collected in [34]. 70% samples of the data set are randomly

Table 3.3: Architecture of CNN model for the second database

Layer	Output dimension
Input	1x500x2
Conv(1,4)	1x500x16
Conv(1,2)	1x500x16
Maxpooling(1,2)	1x250x16
Conv(1,2)	1x250x32
Maxpooling(1,2)	1x125x32
Fully Connect Layer	
Fully Connect Layer	
Fully Connect Layer	
Output	1

selected as the training set, while the remaining 30% samples are the test set. In the SVM model, the RBF kernel is selected, and the penalty parameter C is set as 20, while γ and ϵ are set as 0.1 and 0.1, respectively. In the KNN model, the parameter k is set as 3. In the decision tree model, the depth of the tree is set as 15, and the minimum number of samples required to split an internal node is 20. The parameter of the CNN model can be found in Table 3.3.

The CDF of residual error is shown in Figure 3.10. The unmitigated ranging error has an MAE of 0.21m and a mean error of 0.12m. The SVM can improve the MAE and mean error to 0.16m and 0.05m, respectively. The KNN and decision tree can achieve the MAE of 0.11m and 0.06m, respectively. The mean error for KNN and decision tree are 0.004m and 0.004m. CNN can obtain an MAE of 0.11m and a mean error of -0.03m.

Since the current models perform very well on the locations that have been trained for, it is also essential to explore the performance of the unseen positions. Three positions are selected as unknown positions, and samples collected from these positions are considered as the test set, while the remaining samples are the training set. Figure 3.11 shows the performance of three machine learning models under this circumstance. The MAE of the unmitigated ranging error of this test set is 0.20m, while the mean error is 0.08m. The SVM remains an MAE of 0.19m and a mean error of -0.01m, while the performance of KNN and decision tree are degraded significantly. KNN has a mean error of -0.04m and an MAE of 0.23m. The MAE of the decision tree is 0.37m, and the mean error is -0.22m. Since the result of KNN and decision tree are closely relative to the training samples, and may introduce several estimates with large ranging errors. The CNN model can have a mean error of -0.04m and an MAE of 0.16m. As a result, when the training set is not sufficient enough to cover the map, SVM and CNN are suitable options compared with KNN and decision tree.

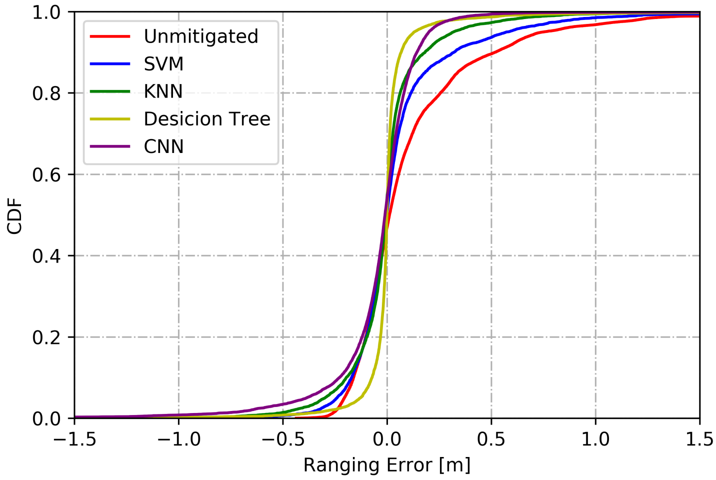


Figure 3.10: CDF of unmitigated ranging error and ranging error mitigated with SVM, KNN, decision tree and CNN of database 2.

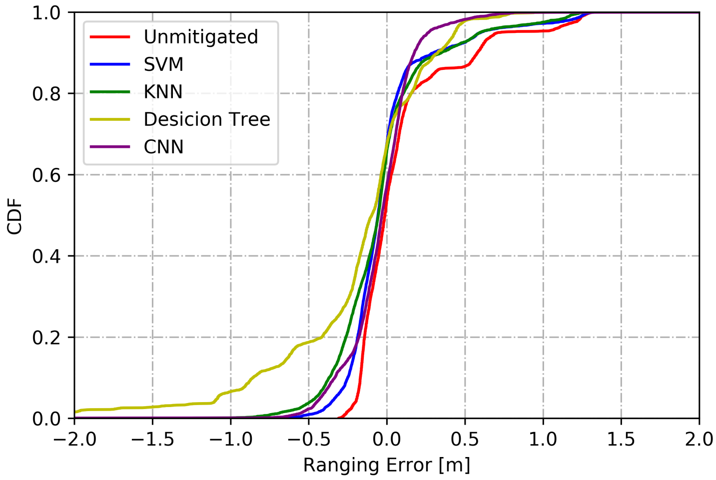


Figure 3.11: CDF of unmitigated ranging error and ranging error mitigated with SVM, KNN and decision tree on three unknown positions of database 2

3.5 Summary

In this chapter, several machine learning algorithms are utilized for ranging error mitigation. We firstly extract ten features from CIR signals, and use these features as input for SVM, KNN and decision tree regression models to estimate the ranging error. Besides manual feature extraction, a CNN model is also proposed, which can estimate the ranging error directly from the CIR signals. These proposed algorithms are trained and tested on two datasets under multiple circumstances. The SVM regression model is proved to have a stable performance of ranging error mitigation in the experiment. The result of SVM will be mainly used in the following chapters of outlier prediction and localization estimation.

Chapter 4

LOS/NLOS Classification

UWB technology can achieve high accuracy under LOS circumstances. In NLOS cases, the ranging accuracy can degrade significantly due to obstruction and reflection. The ranging error mitigation algorithms can improve the ranging estimate performance. However, the average residual ranging error of NLOS cases is still larger than that of LOS cases. By classifying the LOS and NLOS signals, more prior information can be provided for localization algorithms. In the localization process, we can add more weight on LOS ranges compared with NLOS ranges, or ignore the NLOS ranges with large ranging errors to improve the positioning accuracy. In this chapter, we are going to implement a LOS/NLOS classifier and a outlier predictor based on SVM classification model.

4.1 LOS/NLOS Classification

4.1.1 Support Vector Machine for Classification

Support vector machine can solve both the regression problems and the classification problems [42]. Same as the SVM regressor, a SVM classifier can make an optional feature transformation, and transfer the input variables x into $\varphi(\mathbf{x})$. Assume that for a two-class classification problem using linear model, the input data set is $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)$, where $\mathbf{x}_i \in R^n$ and the target value $t_i \in \{-1, +1\}$. The data points are classified according to the linear separation function which is given by

$$y(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \tag{4.1}$$

SVM classifier also introduces a concept of margin, which is defined as the smallest distance between the decision boundary and any of the samples. The separation function should satisfy that $y(\mathbf{x}_i) \geq 1$ for data points having $t_i = 1$ and $y(\mathbf{x}_i) \leq -1$ for data points having $t_i = -1$, which means $t_i * y_i \geq 1$ for all data points. The interval between $y(\mathbf{x}_i) = 1$ and $y(\mathbf{x}_i) = -1$ is the margin. The goal is to learn the location of decision boundary that can separate data points with maximal margin. As a result, the formulation of this optimization problem is given by

$$\begin{aligned} &\underset{w,b}{\arg\min} \quad \frac{1}{2}||\mathbf{w}||^2 \\ s.t. \quad &t_i \cdot y_i \geq 1 \quad i = 1, 2, ...n \end{aligned} \tag{4.2}$$

However, in many cases, it is desirable to allow a few numbers of miss-classified data points to make the model more robust, especially when the database contains noise. The concept of slack variable ξ_i is introduced to allow mistakes. Each data point located inside the margin have a positive value of ξ_i, while the data points located outside the margin have a zero value of ξ_i. The slack variable of each data point is given by

$$\xi_i = \begin{cases} 0, & t_i \cdot y_i > 1 \\ |t_i - y(\mathbf{x}_i)|, & otherwise \end{cases} \tag{4.3}$$

Since the slack variables should not be unnecessarily large, a penalty term should be added to the variable ξ. The formulation of optimization problem can be updated to:

$$\underset{\mathbf{w},b}{\arg\min} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \tag{4.4}$$

where C is the penalty parameter that can control the trade off between allowable penalty and width of the margin. A high value for C indicates that we care more about the errors and expect to classify most of the training data correctly, which may cause a narrow margin or even overfitting problem. The algorithm with a small value for C may be less sensitive to errors and have a wide margin. A suitable value for C should be selected based on the characteristic of the data.

The optimization problem can be transformed into a dual problem, and the regression function can be solved as following:

$$y(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i \kappa(\mathbf{x}, \mathbf{x}_i) + b \tag{4.5}$$

where $\kappa(\mathbf{x}, \mathbf{x}_i)$ is the kernel function. By utilizing kernel function, the input data can be transformed into high dimension without directly calculate the inner product from $\varphi(\mathbf{x})\cdot\varphi(\mathbf{x}')$, and computation amount can be reduced significantly. The kernel functions in SVM classifier are the same as that of SVM regressor. Several common used kernel functions, such as linear kernel, RBF kernel, polynomial kernel and sigmoid kernel, can be utilized. In the LOS/NLOS classifier, radial basis function kernel is selected.

4.1.2 Procedure of LOS/NLOS Classification

1.Data Preparation

Each sample can be labeled as LOS signal or NLOS signal according to the floor plan of measurement environment. The target label is set as 1 for LOS signal, and -1 for NLOS signal. For each CIR sample, ten features can be extracted as described in Section 3.1. The labeled database $\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)\}$ will be used to train and evaluate the SVM classifier, where $\mathbf{x}_i$ is the feature vector for the ith sample, and t_i is the target label. The value of input features should also be normalized to a common scale. In this project, 70% of the samples are randomly selected as the training set, while the remaining 30% samples are the test set. Both the training samples and the test samples can scatter in different positions in the experimental environment.

2. Model Training

We use the training set to fit the classification model. In the fine-tuning process, the 5-fold cross-validation method is utilized to validate the model without using the test data during the training step. After tuning the parameter, we select RBF kernel function for the SVM LOS/NLOS classification model, and the kernel coefficient parameter γ is set to 0.1. The penalty parameter C is set to 10.

3. Model Evaluation

After fitting the classification model with the training set, we can use the test set to evaluate the proposed model. If a data point belongs to the positive class and is correctly labeled, the data point is true positive(TP). Otherwise, if a data point is incorrectly labeled as positive class, it is false positive(FP). If a data point belongs to the negative class and is correctly labeled, the data point is true negative(TN). Otherwise, the data point is false negative(FN). A classifier is evaluated based on several metrics, such as accuracy, precision, recall and f1-score [47]. Accuracy describes the proportion of correct labeled data points, which is given by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

Precision is defined as the fraction of relevant samples among each class, which is given by

$$Precision = \frac{TP}{TP + FP} \tag{4.7}$$

Recall shows the percentage of correct labeled positive predictions among all positive samples, which is given by

$$Recall = \frac{TP}{TP + FN} \tag{4.8}$$

F1-score is a balance between recall and precision, which is given by

$$F1 - score = 2 \times \frac{precision * Recall}{Precision + Recall} \tag{4.9}$$

4.2 Outlier Prediction

The machine learning method proposed in Chapter 3 is proved to be able to mitigate ranging errors effectively. However, about 7% of the data points have residual ranging errors that are significantly larger than the average ranging error. These data points can be considered as outliers and can influence the localization result significantly. In this thesis work, we define the ranges with residual ranging error larger than 1m as outliers. An outlier predictor can be implemented to detect and remove the outliers and improve the localization performance.

The process of outlier detector is shown in Figure 4.1. The input database of outlier detector is $\{(x_1, t_1), (x_2, t_2), ..., (x_N, t_N)\}$, where x_i is the ten features extracted from CIR signals together with the estimated ranging error estimated, and t_i is the outlier label. An SVM classifier is employed for outlier detector. The RBF kernel is selected for kernel function, and kernel coefficient γ is set as 0.1. The regularization parameter C is set as 10.

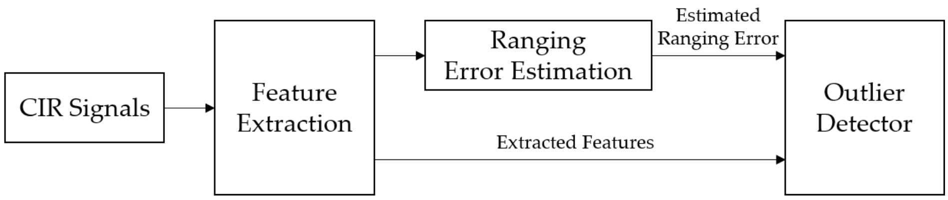


Figure 4.1: Process of outlier detector

4.3 Experimental Result

4.3.1 LOS and NLOS Signals Classification Result

The measurement data from the first database is used to test the LOS/NLOS classifier. In this test, the carrier frequency is 7.5GHz, while the bandwidth of the signal is 1.82GHz. The performance of LOS/NLOS classifier is shown in Table 4.1. The accuracy of SVM classifier can achieve 96%. Figure 4.2 visualize the LOS/NLOS classification results of the data points received by base station 5. Generally, the classifier can achieve a high accuracy and most of the samples can be correctly classified. In this measurement environment, some of the walls are non-load bearing and made of light building materials, so that several NLOS signals with close distances might be incorrectly classified as LOS signals.

We also test the LOS/NLOS classifier under various bandwidth settings. Table 4.2 depicts the accuracy of SVM classifier for UWB signals with different bandwidth. By decreasing the bandwidth from 1.82GHz to 500MHz, the performance of LOS/NLOS classifier is slightly degrade but not influenced too much. Figure 4.3 and Figure 4.4 visualizes the classification results of the data received by base station 5 under different bandwidth. Few wrong predicted

Table 4.1: Performance of LOS/NLOS Classifier

	Precision	Recall	F1-score	Accuracy
NLOS	96%	98%	97%	
LOS	96%	90%	93%	
Total				96%

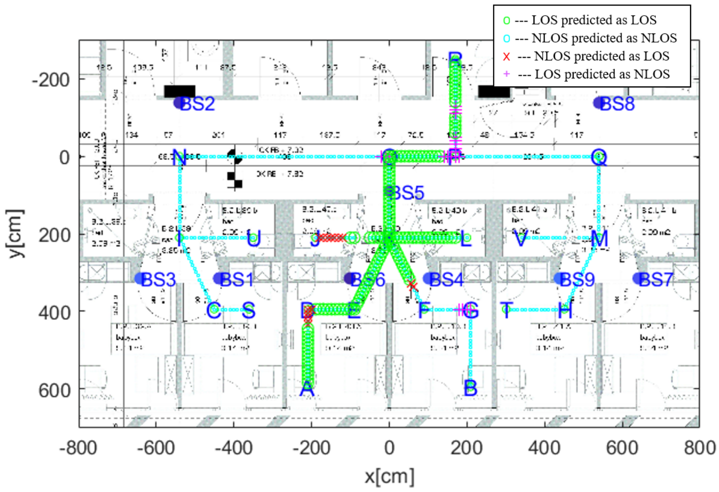


Figure 4.2: Performance of SVM classifier on data collected at base station 5, which is indicated by BS5 in the picture. The carrier frequency is 7.5GHz, and the bandwidth is 1.82GHz.

points are introduced, but the general accuracy is still stable when decreasing the bandwidth.

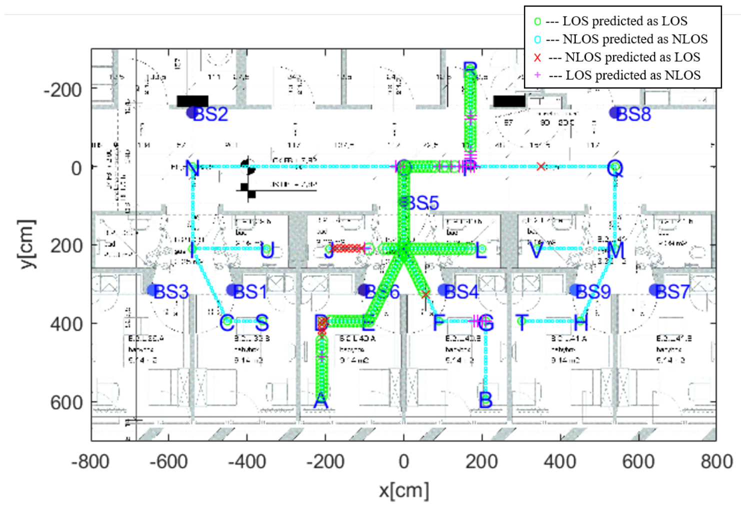


Figure 4.3: Performance of SVM LOS/NLOS classifier on data collected at BS5. The carrier frequency is 7.5GHz, and the bandwidth is 1GHz

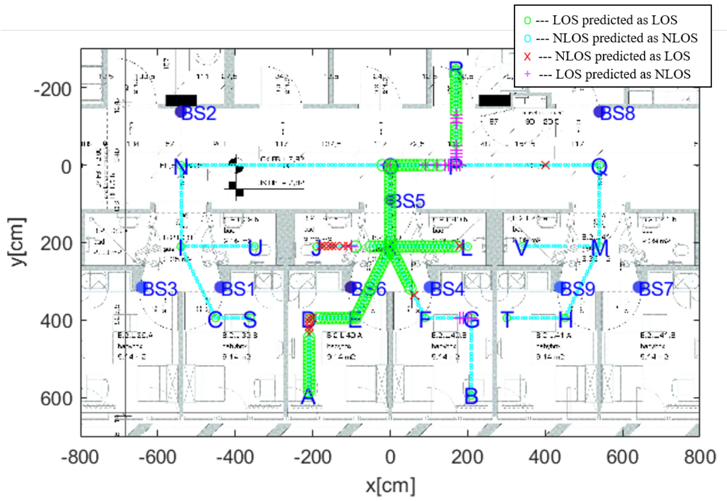


Figure 4.4: Performance of SVM LOS/NLOS classifier on data collected at BS5. The carrier frequency is 7.5GHz, and the bandwidth is 500MHz

Table 4.2: Performance of LOS/NLOS Classifier With Different Bandwidth

Bandwidth	Precision	Recall	F1-score	Accuracy
1.82GHz	96%	98%	97%	96%
1GHz	97%	98%	97%	96%
500MHz	95%	97%	96%	94%

4.3.2 Outlier Detection Results

The outlier detector was tested with the database 1. If the residual ranging error is still larger than 1m after estimating ranging error with regressor, the data point should be labeled as outlier. In this data set, about 7% of the sample points are outliers. The performance of outlier detector is shown in Table 4.3. The accuracy can achieve 99%.

Figure 4.5 shows the CDF of residual ranging errors after removing predicted outliers, where SVM_original is the performance of ranging error mitigated with the SVM regressor, while the CDF of LOS ans NLOS illustrates of the risidual ranging errors of LOS and NLOS respectively after removing the outliers. By removing predicted outliers, the mean absolute ranging error can be reduced from 0.32m to 0.20m, and the proportion of data points with ranging error larger than 1m is decreased significantly, and 90% of the remaining ranging can

Table 4.3: Performance of Outlier Detector

	Precision	Recall	F1-score	Accuracy
Not Outlier	99%	99%	99%	
Outlier	94%	92%	93%	
Total				99%

be reduced to an error smaller than 0.39m. After filtering the outliers, the remaining ranges can be used for the input of localization algorithms.

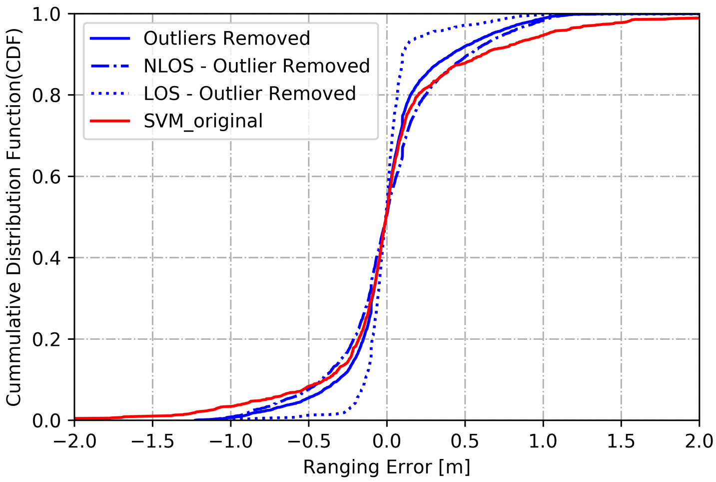


Figure 4.5: CDF of Residual Ranging Errors After Removing Outliers

4.4 Summary

In this chapter, a LOS/NLOS classifier and an outlier predictor are implemented based on the SVM classification model. The models are trained and tested on the database 1. The LOS/NLOS classifier can classify the LOS and NLOS ranges effectively and achieve an accuracy of 96%. While the outlier predictor can identify the outliers with residual ranging error larger than 1m, and achieve an accuracy 99%. The classifiers can provide more prior information for the localization algorithms, which can potentially improve the localization accuracy.

Chapter 5

Localization

In a wireless localization network, there are several anchors located in the known fixed positions, while a mobile agent can move in the network and the position is unknown. As discussed in the previous chapters, the distance between anchor and agent can be measured based on ToA, and ranging errors can be mitigated with machine learning algorithms. In this chapter, we will introduce several single-shot localization algorithms which can estimate the position of the agent based on measured distance and known position of the anchors.

5.1 Linear Least Square Localization

Suppose that there are N anchors located in the known fixed positions in a wireless localization network as in Figure 5.1. The mobile agent is surrounded by N anchors with known positions, and the distances between the agent to each anchor can be measured. With these known values, the position of mobile agent can be calculated with linear least square algorithm [48].

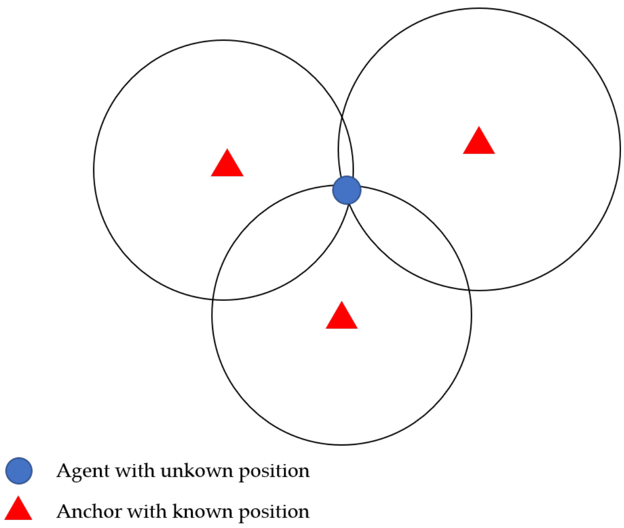


Figure 5.1: Wireless Localization Network

The algorithm can be illustrated in two-dimensional coordinate. The position of the ith ($i \in \{1, 2, ..., N\}$) anchor is denoted as $\mathbf{p}_i$=$[x_i \quad y_i]^T$, while $\mathbf{p}$=$[x \quad y]^T$ is the position of the mobile agent. The measured distance between the mobile agent and the ith anchor is denoted as $\hat{d}_i$, which can be modeled as:

$$\hat{d}_i(\mathbf{p}, \mathbf{p}_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{5.1}$$

A common solution for localization estimation to minimize the sum of the residuals, which is given by:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \quad \sum_{i=1}^{N} (\hat{d}_i - ||\mathbf{p} - \mathbf{p}_i||)^2 \tag{5.2}$$

The optimization problem shown in formula (5.2) is a non-linear least square problem. To solve such a non-linear optimization problem, some numerical algorithms, such as gradient descent or Gauss-Newton algorithm, need to be applied. Another simple solution is to convert the non-linear equations to linear equations, which is defined as linear least square(LLS) algorithm. The formula shown in (5.1) can be dissolved to:

$$\begin{cases} \hat{d}_1^2 = x^2 + y^2 + x_1^2 + y_1^2 - 2xx_1 - 2yy_1 \\ \hat{d}_2^2 = x^2 + y^2 + x_2^2 + y_2^2 - 2xx_2 - 2yy_2 \\ ... \\ \hat{d}_N^2 = x^2 + y^2 + x_N^2 + y_N^2 - 2xx_N - 2yy_N \end{cases} \tag{5.3}$$

These equations contain non-linear parts x^2 and y^2, which can be eliminated by subtracting the first equation from the rest of equations, and the set of equations are converted to:

$$\begin{cases} \hat{d}_2^2 - \hat{d}_1^2 = (x_2^2 - x_1^2) + (y_2^2 - y_1^2) - 2(x_2 - x_1)x - 2(y_2 - y_1)y \\ \hat{d}_3^2 - \hat{d}_1^2 = (x_3^2 - x_1^2) + (y_3^2 - y_1^2) - 2(x_3 - x_1)x - 2(y_3 - y_1)y \\ ... \\ \hat{d}_N^2 - \hat{d}_1^2 = (x_N^2 - x_1^2) + (y_N^2 - y_1^2) - 2(x_N - x_1)x - 2(y_N - y_1)y \end{cases} \tag{5.4}$$

The problem can be rewritten into the following linear matrix expression:

$$\mathbf{Ap} = \mathbf{b} \tag{5.5}$$

where the matrix $\mathbf{A}$ and $\mathbf{b}$ are given by:

$$\mathbf{A} = 2 \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ ... & ... \\ x_N - x_1 & y_N - y_1 \end{bmatrix} \tag{5.6}$$

$$\mathbf{b} = \begin{bmatrix} d_1^2 - d_2^2 + (x_2^2 - x_1^2) + (y_2^2 - y_1^2) \\ d_1^2 - d_3^2 + (x_3^2 - x_1^2) + (y_3^2 - y_1^2) \\ ... \\ d_1^2 - d_N^2 + (x_N^2 - x_1^2) + (y_N^2 - y_1^2) \end{bmatrix} \tag{5.7}$$

The matrix $\mathbf{A}$ and $\mathbf{b}$ can be calculated with the known information. From formula (5.5), we can get the estimated linear least square position of the mobile agent, which is given by:

$$\hat{\mathbf{p}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{5.8}$$

5.2 Reference Selection for Linear Least Square Localization

In the previous linear least square solution, we select the first anchor as the reference when eliminating the non-linear elements. When computing the matrix $\mathbf{b}$, as shown in formula(5.7), all the elements are relevant to the estimated distance between the mobile agent and the first anchor. If the estimated distance d_1 contains a large ranging error, the estimated position can be noisy and the localization accuracy can be degraded. As a result, it is critical to apply a reference selection strategy to improve localization accuracy, which can be defined as linear least square with reference selection(LLS-RS) algorithm.

A simple reference selection strategy is proposed in [49]. If the mobile agent is close to an anchor, the estimated range tends to contain a small ranging error. As a result, the anchor with the minimum estimated distance among all the distance measurements can be selected as the reference anchor. The index of the reference anchor is given by:

$$r = \arg\min_i \quad \hat{d}_i \tag{5.9}$$

After selecting the reference anchor according to above selection rule, the matrix $\mathbf{A}$ and $\mathbf{b}$ can be obtained as formula(5.10) and (5.11), and the resulting position of mobile agent can contain less bias.

$$\mathbf{A} = 2\begin{bmatrix} x_1 - x_r & y_1 - y_r \\ x_2 - x_r & y_2 - y_r \\ \cdots & \cdots \\ x_{r-1} - x_r & y_{r-1} - y_r \\ x_{r+1} - x_r & y_{r+1} - y_r \\ \cdots & \cdots \\ x_N - x_r & y_N - y_r \end{bmatrix} \tag{5.10}$$

$$\mathbf{b} = \begin{bmatrix} d_r^2 - d_1^2 + (x_1^2 - x_r^2) + (y_1^2 - y_r^2) \\ d_r^2 - d_2^2 + (x_2^2 - x_r^2) + (y_2^2 - y_r^2) \\ \cdots \\ d_r^2 - d_{r-1}^2 + (x_{r-1}^2 - x_r^2) + (y_{r-1}^2 - y_r^2) \\ d_r^2 - d_{r+1}^2 + (x_{r+1}^2 - x_r^2) + (y_{r+1}^2 - y_r^2) \\ \cdots \\ d_r^2 - d_N^2 + (x_N^2 - x_r^2) + (y_N^2 - y_r^2) \end{bmatrix} \tag{5.11}$$

5.3 Weighted Linear Least Square

Linear least square localization algorithm allows us to estimate the position of the agent without prior knowledge. However, LLS localization algorithm gives the same weight to each signal, regardless of the ranging error of each estimated distance. To improve the localization accuracy, we can assign adaptive weight to each signal according to the residual ranging error. By utilizing weighted linear least square(WLLS) algorithm, the estimated distance with higher accuracy can contribute more to the estimate of position.The optimization problem can be updated to:

$$\mathbf{p} = \sum_{i=1}^{N} \beta_i(\hat{d}_i - ||\mathbf{p} - \mathbf{p}_i||)^2 \tag{5.12}$$

where β_i is the weight for the ith anchor, and can reflect the reliability of the estimated distance from the ith anchor to the mobile agent.

It is essential to select the appropriate weight β_i for each anchor. The NLOS ranges should be assigned a smaller weight compared with LOS ranges. Besides, the ranges with larger distance tends to contain larger ranging errors, so the distance should also be considered for the weight matrix. According to [50], we can follow the best linear unbiased estimator to determine the optimum weight β_i, which can be given by:

$$\beta_i = \frac{1}{4} \cdot \frac{1}{\sigma_i^2 d_i^2} \tag{5.13}$$

where σ_i^2 is the variance of the residual ranging errors. If the signal is classified as the LOS signal, σ_i^2 is equal to σ_{LOS}^2. Otherwise, σ_i^2 is equal to σ_{NLOS}^2. σ_{LOS}^2 and σ_{NLOS}^2 can be obtained with the known measurements in practice. The true distance may be not available in practice, so the estimated distance can be used for calculating the weight matrix.

The non-linear optimization problem can be converted to linear problem as discussed in the previous chapter. The weighted linear least square estimate can be solved as:

$$\hat{\mathbf{p}} = (\mathbf{A}^T\mathbf{w}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{w}\mathbf{b} \tag{5.14}$$

where matrix $\mathbf{A}$ and $\mathbf{b}$ is the same as (5.10) and (5.11), and the weighted matrix $\mathbf{w}$ is a diagonal matrix of size N-1, which is given by:

$$\mathbf{w} = diag(\beta_1, \beta_2, ..., \beta_{r-1}, \beta_{r+1}, ..., \beta_N) \tag{5.15}$$

where the diag-operator is to create a square diagonal matrix with all the following elements on the main diagonal.

5.4 Localization with Support Vector Machine

Least square localization algorithm utilizes known position of the anchor and estimated distances to build a model and estimate the position of mobile agent. Since the machine learning algorithm is also exploring the mapping function from the input features to the output data, it is reasonable to employ machine learning algorithm for localization. In this work, we apply support vector machine regressor for localization algorithm.

The position of anchors are fixed, and can be considered as constant values, which will not contribute to the SVM model. As a result, estimated distances between the nodes $\{d_1, d_2, ..., d_N\}$ are considered as the input feature vectors for support vector machine.

A SVM regression model is employed for each dimension, and two SVMs are used in total. For x dimension, the training data set is $\{[d_1, d_2, ..., d_N]$, x$\}$. For y dimension, the training data set is $\{[d_1, d_2, ..., d_N]$, y$\}$.

The model is trained and validated with the 5-fold cross validation method, and the parameters are set to the suitable value. Radial basis function kernel is selected for two SVM regression models. Kernel coefficient parameter γ is set to 0.1. The regularization parameter C is set to 10, while the parameter ϵ for ϵ-intensive error function is set to 0.1.

5.5 Localization Results

A mobile agent can move in the wireless localization network, and the UWB signals from agent to mobile anchor can be LOS or NLOS signals. By utilizing the algorithms introduced in Chapter 3, the ranging error can be mitigated effectively. The classification algorithm introduced in Chapter 4 can provide LOS/NLOS classification and outlier detection result, which can be used as prior information for location estimate. With these known information, the position of the mobile agent can be computed based on the localization algorithms proposed in chapter 5. In this section, we are going to test the proposed localization algorithms with the first database. The localization algorithms are tested based on the ranges whose ranging errors are mitigated with SVM.

5.5.1 Linear Least Square Localization Results

The outlier detector can identify the measurement outlier whose ranging error is larger than 1 meter. Since only 5% of the estimated ranges are detected as outliers, the anchor information is still sufficient for applying localization algorithms after removing the outlier anchors. The information of the anchor which is identified as outlier will not contribute to the localization result, and only the remaining anchors whose ranging error is smaller than 1m will be considered in LLS localization algorithm.

Figure 5.2 visualizes the performance of linear least square localization. The average localization error is 51.79cm. In this implementation, BS 1 is selected as the fixed reference anchor. It is obvious that the estimate positions on the right side have larger ranging error compared with that on the left side. There are several walls located in the wireless line between agents on the right side and BS 1, and most of those signals are considered as NLOS signals, which have larger ranging errors. Selecting anchor 1 as the fixed reference anchor can introduce more localization bias for the nodes far away from the reference anchor.

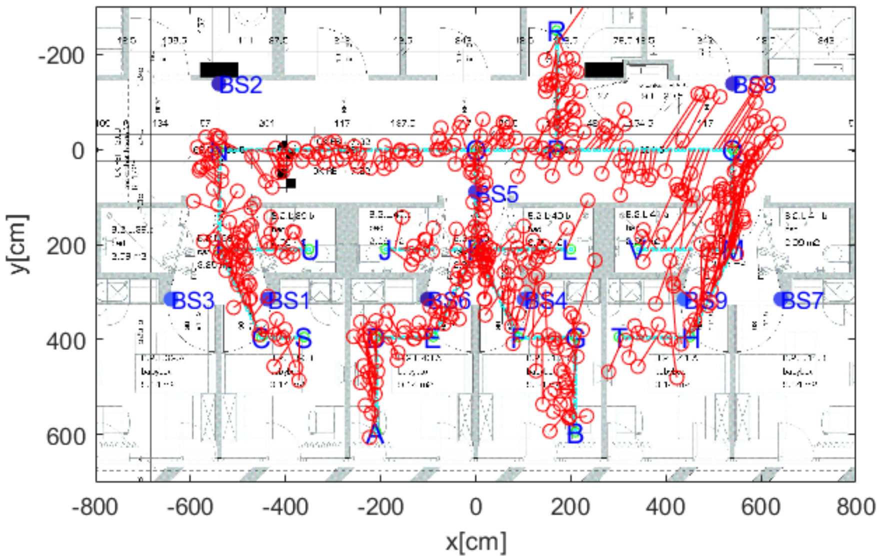


Figure 5.2: Localization performance of linear least square algorithm

5.5.2 Linear Least Square Localization with Reference Selection

Selecting a fixed reference anchor can introduce bias ranging errors. Reference selection is proposed as an improvement strategy for traditional LLS localization algorithm. For each node, the anchor with the shortest estimated range is selected as the reference anchor. The performance for LLS with reference selection is depicted on Figure 5.3. By utilizing LLS-RS, a mean absolute localization error of 34.21cm can be obtained. Compared with the performance of LLS, the results of LLS-RS contain fewer outliers.

5.5.3 Weighted Linear Least Square Localization Results

Based on the LLS-RS algorithm, we can add suitable weight for each range, so that the ranges with smaller accuracy can make more contribute to the localization result, and the localization accuracy can be improved. The value of the weight can be calculated with formula (5.13). The value of σ_{LOS}^2 and σ_{NLOS}^2 can be calculated with the known measurements and estimated distances. In this experiment, the value of σ_{LOS}^2 is 6 cm^2, while σ_{NLOS}^2 is 48 cm^2. The localization performance of WLLS localization algorithm is shown in Figure 5.4. The mean absolute localization error is 24.56cm for WLLS algorithm. A significant improvement can be achieved by utilizing weight strategy.

5.5.4 Localization with SVM

The position of the mobile agent can be obtained with SVM. The 70% of the positions can be randomly selected as training set, while the remaining 30% are the test set. Two SVMs are

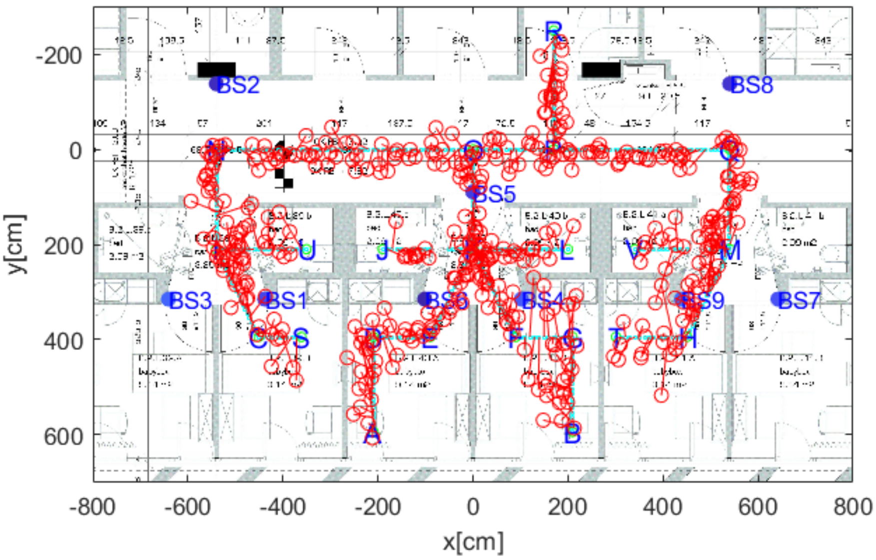


Figure 5.3: Localization performance of linear least square algorithm with reference selection

implemented for each dimension. The training set is used to fine-tuning the hyperparameter and train the model. The test set can be utilized as the input of the trained model to test the performance of the proposed algorithm. In this work, polynomial kernel is used in the SVMs, and the degree is set as 3. The kernel coefficient parameter γ is set as 0.1, while the penalty parameter C and the parameter ϵ are set as 5 and 0.1, respectively. The performance of SVM localization algorithms is shown on Figure 5.5. The overall MAE can achieve 19.66m. The MAE of the training set can be 16.04cm, while the MAE of the test set is 28.08cm. 90% of the estimated positions in the testing set can have a localization error smaller than 54.82cm.

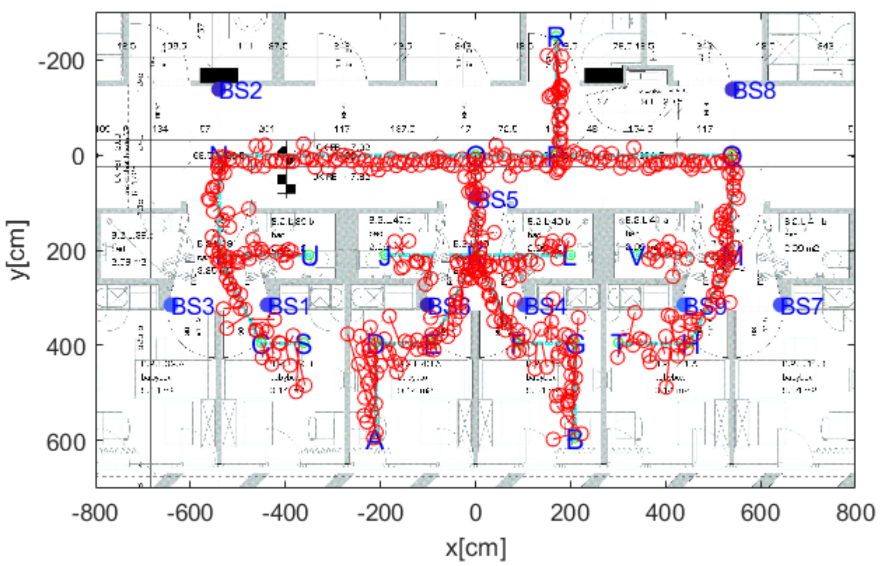


Figure 5.4: Localization Performance of Weighted Linear Least Square Algorithm

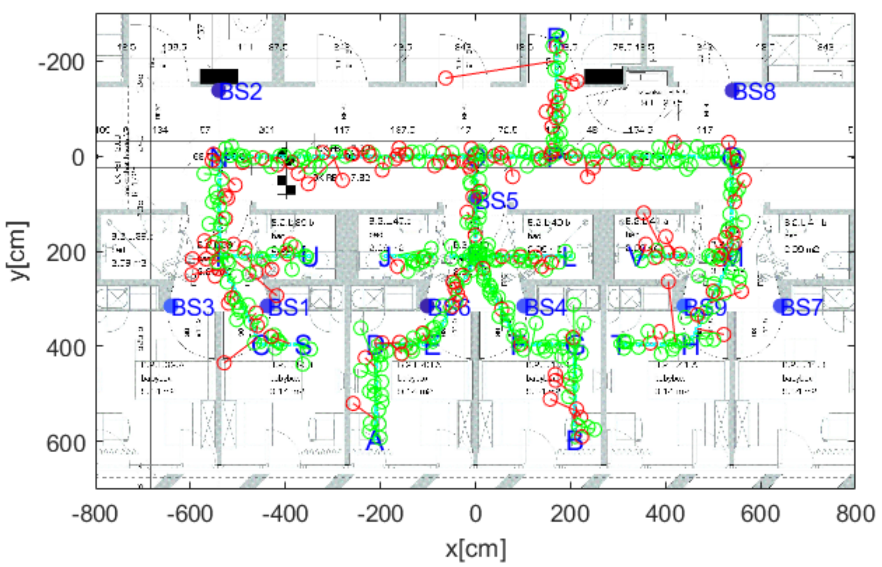


Figure 5.5: Localization performance of SVM localization algorithm. The green points are the estimated position of the training set, while the red points are the estimated position of the test set.

5.6 Summary

In this chapter, four localization algorithms were introduced, and the methods were tested on the database 1. The LLS algorithm can provide a linear solution for localization estimation, but the localization errors are significant in several positions due to selecting the fixed reference base station, and can only achieve an average localization accuracy of 51.79cm. The LLS-RS algorithm can provide a reference anchor selection strategy, improving the localization accuracy to 34.21cm. The WLLS algorithm can assign different weights to each range, and the ranges with small errors can contribute more to the localization estimation. The localization accuracy can be improved to 24.56cm with the WLLS algorithm. Finally, SVM regression models were employed for localization estimation. The SVM localization algorithm can obtain an overall MAE of 19.66cm, while the MAE of 16.04cm on the training set and 28.08cm on the test set. Both the SVM and WLLS with reference selection algorithm can provide an accurate estimate of the localization compared with traditional linear least square algorithm.

Chapter 6

Conclusions

In this thesis work, we design a high-accurate UWB localization system, and the research questions proposed in Chapter 1 are solved as following:

1. What information can we extract from received UWB pulse shape signals to improve indoor localization accuracy?
Ten features can be extracted from the UWB CIR signals, which are energy, maximum amplitude, standard deviation, kurtosis, rician K factor, signal to noise ratio, mean excess delay, RMS delay spread, rise time, and estimated distance. These features can be used as input of machine learning algorithms, such as SVM, KNN and decision tree, to estimate the ranging errors. The combination of these ten features are proved to obtain the best performance compared with other combinations.

2. Which machine learning algorithm can we use to train the UWB wireless localization link and induce the range and localization errors?
In this work, machine learning algorithms such as SVM, KNN and decision tree are trained and tested. CNN is also considered to mitigate the ranging error without manual feature extraction. These algorithms are trained and tested in two different environments. One of the data sets is collected in an indoor corridor environment, which the other is collected in a complex warehouse with metal racks. The proposed algorithms are proved to be able to mitigate ranging errors effectively in different environments and various circumstances. The unmitigated MAE of the database 1 is 1.90m. This error can be reduced to 0.32m with the SVM model. In the database 2, the SVM can reduce the MAE of the ranging error from 0.21m to 0.16m. The SVM model can present stable and high accurate performance in both when the test set is scattered randomly in the experimental environment or the test set is in unknown routes.circumstances when the data set is sufficient to cover the whole map. As a result, SVM can be selected as the optimal ranging error mitigation solution in this work.

Besides, a LOS/NLOS classifier and an outlier predictor are also employed based on the SVM classifier. The LOS/NLOS classifier can achieve an accuracy of 97%, while the outlier predictor can predict the outlier, which has a ranging error larger than 1m, with an accuracy of 99%. In the database 1, the MAE of the ranging error can be mitigated to 0.20m by removing the outliers with outlier predictor. These models can identify the ranges with large ranging errors and provide additional prior knowledge for localization algorithms.

Several localization algorithms are also implemented to realize 2D indoor localization. The LLS can provide an average localization accuracy of 51.79cm, while the LLS-RS introduces a reference selection strategy, and can achieve an accuracy of 34.21cm. The WLLS algorithm can assign different weight to each range, and can improve the accuracy to 24.56cm. Besides the traditional linear least square algorithms, SVM regression model is also employed in a localization algorithm. The overall accuracy can be 19.66cm, while the MAE for the testing set is 28.08cm. Both the SVM and WLLS with reference selection algorithm can provide an accurate estimate of the localization.

Since the current methods are mainly based on off-line computation, some technologies such as edge computing should also be considered and employed for real-time position estimation applications in future work.

Bibliography

[1] Lichtenegger H. Collins J. Hofmann-Wellenhof, B. *Global Positioning System: Theory and Practice.* Springer-Verlag Wien, 2001.

[2] A. El-Rabbany. *Introduction to GPS: The Global Positioning System, Second Edition.* 2006.

[3] J. Marek and L. Štěpánek. Accuracy and availability of the satellite navigation system gps. pages 121–124, April 2010.

[4] S. K. Moore. Super-accurate gps coming to smartphones in 2018. *IEEE Spectrum*, 54(11):10–11, 2017.

[5] Andreas Möller, Matthias Kranz, Robert Huitl, Stefan Diewald, and Luis Roalter. A mobile indoor navigation system interface adapted to vision-based localization. 2012.

[6] Matteo Ridolfi, Stef Vandermeeren, Jense Defraye, Heidi Steendam, Joeri Gerlo, Dirk Clercq, Jeroen Hoebeke, and Eli De Poorter. Experimental evaluation of uwb indoor positioning for sport postures. *Sensors*, 18:168, 01 2018.

[7] T. Feigl, C. Mutschler, and M. Philippsen. Supervised learning for yaw orientation estimation. pages 206–212, 2018.

[8] D. Roth, C. Klelnbeck, T. Feigl, C. Mutschler, and M. E. Latoschik. Beyond replication: Augmenting social behaviors in multi-user virtual realities. pages 215–222, 2018.

[9] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan. Received-signal-strength-based indoor positioning using compressive sensing. *IEEE Transactions on Mobile Computing*, 11(12):1983–1993, 2012.

[10] A.W.Reza and T.K. Geok. Investigation of indoor location sensing via rfid reader network utilizing grid covering algorithm. *Wireless Pers Commun*, pages 67–80, 2009.

[11] L. Mainetti, L. Patrono, and I. Sergi. A survey on indoor positioning systems. pages 111–120, 2014.

[12] D. Yan, B. Kang, H. Zhong, and R. Wang. Research on positioning system based on zigbee communication. pages 1027–1030, 2018.

[13] L. Chen, E. H. Wu, M. Jin, and G. Chen. Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation. *IEEE Sensors Journal*, 14(11):4034–4042, 2014.

[14] N. Arrue, M. Losada, L. Zamora-Cadenas, A. Jiménez-Irastorza, and I. Vélez. Design of an ir-uwb indoor localization system based on a novel rtt ranging estimator. pages 52–57, 2010.

[15] M. Malajner, P. Planinšič, and D. Gleich. Uwb ranging accuracy. pages 61–64, 2015.

[16] Liuqing Yang and G. B. Giannakis. Ultra-wideband communications: an idea whose time has come. *IEEE Signal Processing Magazine*, 21(6):26–54, 2004.

[17] Concettina Buccella, Mauro Feliziani, and Giuliano Manzi. Penetration of ultra-wideband (uwb) communication signals through walls. *Ultra-Wideband, Short-Pulse Electromagnetics 7, ISBN 978-0-387-37728-5. Springer-Verlag New York, 2007, p. 784*, 01 2007.

[18] G. Durisi and S. Benedetto. Performance evaluation of th-ppm uwb systems in the presence of multiuser interference. *IEEE Communications Letters*, 7(5):224–226, 2003.

[19] J. Tiemann, F. Eckermann, and C. Wietfeld. Multi-user interference and wireless clock synchronization in tdoa-based uwb localization. *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6, 2016.

[20] D. Dardari, C. Chong, and M. Win. Threshold-based time-of-arrival estimators in uwb dense multipath channels. *IEEE Transactions on Communications*, 56(8):1366–1378, 2008.

[21] V. Djaja-Josko and J. Kolakowski. A new method for wireless synchronization and tdoa error reduction in uwb positioning system. pages 1–4, 2016.

[22] M. Laaraiedh, N. Amiot, and B. Uguen. Refined characterization of rssi with practical implications for indoor positioning. pages 1–5, 2013.

[23] Z. Tan, X. Zhu, Z. Zhao, B. Liu, Z. Zhu, M. Li, and Z. Nie. Uwb-aoa estimation method based on a spare antenna array with virtual element. pages 1–3, 2018.

[24] Chien-Sheng Chen. Artificial neural network for location estimation in wireless communication systems. *Sensors (Basel, Switzerland)*, 12:2798–817, 12 2012.

[25] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen. Comparison of hybrid localization schemes using rssi, toa, and tdoa. pages 1–5, 2011.

[26] F. Benedetto, G. Giunta, A. Toscano, and L. Vegni. Dynamic los/nlos statistical discrimination of wireless mobile channels. pages 3071–3075, 2007.

[27] V. Barral, C. J. Escudero, and J. A. García-Naya. Nlos classification based on rss and ranging statistics obtained from low-cost uwb devices. pages 1–5, 2019.

[28] Z. Zeng, S. Liu, and L. Wang. Nlos identification for uwb based on channel impulse response. pages 1–6, 2018.

[29] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo. Uwb nlos/los classification using deep learning method. *IEEE Communications Letters*, pages 1–1, 2020.

[30] J. Choi, W. Lee, J. Lee, J. Lee, and S. Kim. Deep learning based nlos identification with commodity wlan devices. *IEEE Transactions on Vehicular Technology*, 67(4):3295–3303, 2018.

[31] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win. A machine learning approach to ranging error mitigation for uwb localization. *IEEE Transactions on Communications*, 60(6):1719–1728, 2012.

[32] R. Zandian and U. Witkowski. Nlos detection and mitigation in differential localization topologies based on uwb devices. pages 1–8, 2018.

[33] Q. Zhang, D. Zhao, S. Zuo, T. Zhang, and D. Ma. A low complexity nlos error mitigation method in uwb localization. pages 1–5, 2015.

[34] J. Fontaine, M. Ridolfi, B. Van Herbruggen, A. Shahid, and E. De Poorter. Edge inference for uwb ranging error correction using autoencoders. *IEEE Access*, 8:139143–139155, 2020.

[35] Valentín Barral, Carlos Escudero, José García-Naya, and Roberto Maneiro-Catoira. Nlos identification and mitigation using low-cost uwb devices. *Sensors*, 19:3464, 08 2019.

[36] J. Tiemann, F. Eckermann, and C. Wietfeld. Fcc first report and order: In the matter of revision of part 15 of the commission's rules regarding ultra-wideband transmission systems. *FCC*, pages 01–48, 2002.

[37] Anu A. Gokhale. *Introduction to Telecommunications (2nd ed.)*. Thomson Delmar Learning, 2004.

[38] J. Romme, J. H. C. van den Heuvel, G. Dolmans, G. Selimis, K. Philips, and H. de Groot. Measurement and analysis of uwb radio channel for indoor localization in a hospital environment. *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*, pages 274–279, 2014.

[39] IDLab. *Industrial IoT Lab*. Accessed:Jun.2020.[Online].Available:https://www.ugent.be/ea/idlab/en/research/research-infrastructure/industrialiot-lab.htm.

[40] Ben Van Herbruggen, Bart Jooris, Jen Rossey, Matteo Ridolfi, Nicola Macoir, Quinten Van den Brande, Sam Lemey, and Eli De Poorter. Wi-pos: A low-cost, open source ultra-wideband (uwb) hardware platform with long range sub-ghz backbone. *Sensors*, 19(7), 2019.

[41] N. Shroff and K. Giridhar. Biased estimation of rician k factor. pages 1–5, 2007.

[42] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] Vojislav Kecman. Support vector machines – an introduction. *Support Vector Machines: Theory and Applications*, 177:605–605, 05 2005.

[44] P.M. Pardalos A. Mucherino, P.J. Papajorgji. *Data Mining in Agriculture*. Springer, New York, NY, 2009.

[45] Chantal D. Larose Daniel T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley Sons, Ltd, 2014.

[46] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

[47] David Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness correlation. *Mach. Learn. Technol.*, 2, 01 2008.

[48] Y. Wang. Linear least squares localization in sensor networks. *J Wireless Com Network*, 51, 2015.

[49] I. Guvenc, S. Gezici, F. Watanabe, and H. Inamura. Enhancements to linear least squares localization through reference selection and ml estimation. pages 284–289, 2008.

[50] RM Buehrer SA Zekavat. *Handbook of position location: theory, practice, and advances.* John Wiley Sons, 2011.