AN ABSTRACT OF THE DISSERTATION OF

Zhenqiang Su for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on May 31, 2017.

Title: Position Estimation in Indoor Localization with Trilateration

Abstract approved: ____

Huaping Liu Professor

Triangulateration uses geometric distance to estimate the location of user by employing techniques like received signal strength (RSS), time-of-arrival (TOA), time-difference-of-arrival (TDOA) and image processing.

Radio frequency (RF) signal positioning using TOA or TDOA techniques generally requires timing synchronization of the anchors and/or the anchors and targets. If the desired position accuracy is high and coverage area is large, timing synchronization will be an extremely challenging issue. The first part of this dissertation focuses on improving the performance or deployability of indoor localization systems. Specifically, we propose a synchronization-free positioning network architecture that eliminates the need of timing synchronization. Another problem that remains unsolved in RF based localization is the non-line-of-sight (NLOS) problem, which greatly degrades the positioning performance. We propose a semidefinite programing (SDP) with a soft-minimal method and an NLOS link identification method with bias deduction to mitigate the NLOS error TOA systems. For TDOA systems, NLOS mitigation is more difficult since a reference should be fixed first. To overcome this problem, we propose a method to transform the TDOA architecture into a TOA one, and then form a SDP problem with new constraints.

To avoid the special problems and difficulties in RF signal positioning, such as the synchronization and NLOS problems, in the second part of the dissertation, we propose an image-tag based localization using image processing and convolutional neural network (CNN). In the proposed method, after the segmentation of the tag from the image, information such as the tag ID, the distance, and the angle with reference to the camera are retrieved through CNNs. The camera position is finally reliably and accurately estimated from such retrieved information. The proposed method simplifies the system and provide good accuracy compare to RF based system. In addition, the proposed method effectively resolve those issues that exist in the traditional image-based localization, like the high cost, blind spot problems and unreliable and not scalable for in changing environments.

Position Estimation in Indoor Localization with Trilateration

by

Zhenqiang Su

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented May 31, 2017
Commencement June 2017

Doctor of Philosophy dissertation of Zhenqiang Su presented on May 31, 2017.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical and Computer Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Zhenqiang Su, Author

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

Chapter 1: Introduction

1.1 Indoor positioning

Location based services (LBSs) are a significant permissive technology that are becoming a vital part of life. The global positioning system (GPS) [1] has been widely used for outdoor localization and navigation and has already become an inevitable part of life. However, the GPS generally is not suitable to establish indoor locations, due to the signal attenuation caused by building construction materials. Therefore, to provide a LBS that can work in both outdoor and indoor scenarios, an indoor positioning system (IPS) should be built. IPS can be used for many applications.

To name a few, one can consider the location detection of products stored in a warehouse, location detection of medical personnel or equipment in a hospital, location detection of firemen in a building on fire, detecting the location of police dogs trained to find explosives in a building, and finding tagged maintenance tools and equipment scattered within a factory.

The primary progress in IPS has just been made during the last ten years or so. Therefore, both the research and commercial products in this area are new, and many researchers in academia and industry are currently involved in the research and development of these systems with different kinds of technologies. Among these technologies, however, it has by now become apparent that there is no overall solution based on a single solution, such as that provided outdoors by satellite-based navigation. We are still far from achieving cheap, simple, and efficient provision of global indoor positioning with an accuracy of around or less than 1 meter. In this section, the currently available technologies are reviewed in two dimensions: the resource/infrastructures and the technologies. The resources/infrastructures refer to the physically visible or existing objects, like satellite and images. Technologies refer to the methods used to identify the location, like timing based methods.

As shown in the list below (it's not a complete list and it's still growing), many kinds of resources/infrastructures are examined and studied in the published papers for the possible candidates of IPS.

- **Satellites**: Like GPS, it's also used in the IPS but with poor accuracy.
- **WiFi/Bluetooth/ZigBee/Cells infrastructures**: The IPS could be achieved through the signal strength or the timing information related to the received signal [2,3].
- **Camera**: With the images or videos from the camera, the location is estimated using matching or estimation [4].
- **Magnetic sensors**: Using the magnetic sensor in a mobile device, the location is estimated through the received magnetic waves from the earth, just like a bird senses its position through an inner magnetic analysis system [5].
- **Light**: The location of a user can be identified by differentiating the light source of received light strength [6].
- **Inertial sensors**: Motion sensors (accelerometers) and rotation sensors (gyroscopes) would continuously calculate via dead reckoning the position, orientation, and velocity of a moving object [7].
- **Ultrasonic wave**: It uses the received signal strength or timing information of the audio signal to estimate the location of a user.
- **Combination of above resources**: To achieve better accuracy and robustness, different kinds of resources can be combined at the cost of increasing complexity.

From the above list, the readers can get a general idea about what different kinds of things are used to achieve an IPS. However, there is currently no answer about what resource is a better candidate for IPS, as each presents its own advantages and disadvantages.

With the different kinds of resources, different kinds of methods and techniques are used to estimate/identify the location of the user.

- **Timing-based methods**: The typical examples are time-of-arrival (TOA) [8] and time-difference-of-arrival (TDOA) [9], as shown in Figs. 1.1 and 1.2, respectively. Both of these exploy the time of flight of the signal from transmitter to the receiver in order to calculate the distance related information of the receiver in reference to the transmitter. This distance information can be used to esimate the location of the user. The GPS is a typical example using this method.

- **Angle-based methods**: Like angle-of-arrival (AOA), as shown in Fig. 1.3, and angle-of-departure (DOA), angle-based methods try to determine the relative direction of the user in reference to the fixed anchor, like a transmitter array [10].

- **Signal strength**: As a signal, which includes the normal radio wave and the lights, will degrade along the transmission line, so does the attenuation of the received signal reflect how long it has travelled from transmitter. Therefore, the distance between the user and the transmitter can be estimated. With several such distances, the localization could be achieved. The typical resources used here are WiFi/Bluetooth/ZigBee signal [11].

- **Location awareness**: Each wireless node, like the WiFi access point (AP) or the Bluetooth node, has limited coverage. The mobile device can only communicate with these nodes within their range. Therefore, the location of the user could be known to be within the coverage range of a node when the user is communicating with this node. Apparently, the accuracy depends on the coverage of the node. The typical resources used here are WiFi, Blutooth and ZigBee nodes.

- **Matching**: The matching method is usually used in image or video related localization. The basic idea behind image-based localization is that a database of structured images of the indoor environment is constructed by using image matching and the bundle adjustment algorithm. Then each image's relative pose, like its position and orientation, is estimated and the semantic locations of images are tagged. The location of the user can then be determined by comparing a photo taken by the mobile phone to the image in the database [12].
- **Fingerprint**: The system relies on the recording of the signal strength from several transmitter nodes within range and stores this information in a database along with the known coordinates of the client's device in an offline phase. During the online tracking phase, the current measured signal strength from all the available transmitters is compared to those stored in the fingerprint and the closest match is returned as the estimated user location. Typical resource used is Wi-Fi APs [13].
- **Combination of the above methods**

Although there are many different kinds of possible methods to achieve IPS, there are no methods that are dominant in IPS, as each one has advantages over the rest. To achieve better performance, several different kinds of resources and methods may be used, like the IPS using Wi-Fi APs and inertial sensors as the resource and employing the received signal strength or timing-based methods to estimate the location.

1.2 Indoor positioning with trilateration

Trilateration uses geometric distance to obtain the location of a user or target. In radio frequency (RF) signal-related indoor localization, the distance related infor-

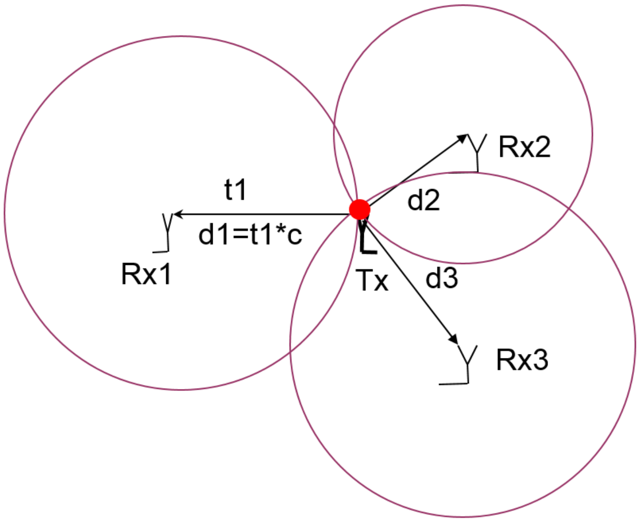


Figure 1.1: TOA method.

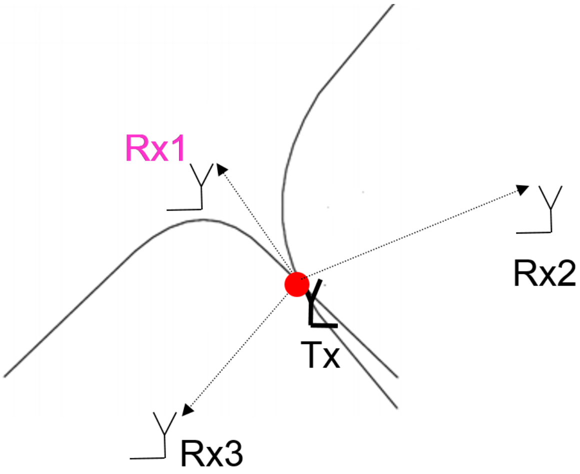


Figure 1.2: TDOA method.

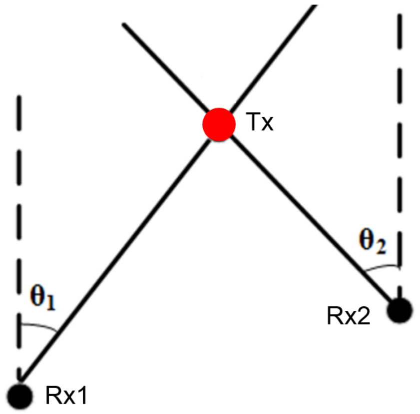


Figure 1.3: AOA method.

mation is usually obtained through measuring the signal traveling time from the transmitter to the receiver. Then the distance information is used to estimate the location of the user or target through trilateration. In this case, techniques like TOA and TDOA are usually used. Another method in RF based localization system to estimate such distance is to measure the received signal intensity. However, the accuracy of the obtained distance is usually low as the signal intensity can be affected by many factors, like signal reflection and NLOS effects.

The distance could also be obtained through the image processing method.

In this section, several techniques related to trilateration that will be used in this dissertation are discussed, like the TOA, the TDOA, and image processing. In addition, AOA is also introduced. Although AOA is not a trilateration method, it's used in the final chapter.

1.2.1 Time-of-arrival (TOA)

TOA is the travel time of a radio signal from a single transmitter to a remote single receiver. As shown in Fig. 1.1, there are three receivers (Rx1, Rx2, and Rx3) and a transmitter, which is also the target to be localized. When the Tx emits a signal at a known time instant, all three receivers will receive the signal at some time instant. Therefore, the time of flight from Tx to Rx could be estimated. For example, the time of flight from Tx to Rx1 is t1. And the distance between Tx and Rx1 is known through $d1 = t1 * c$, where c is the speed of light. With Rx1 as the center and $d1$ as the radium, a circle could be formed and the target should be on the circle. With the same method, two other circles could be formed for Rx2 and Rx3, and the common intersection of the three circles will identify the location of the target.

In order to calculate the time of flight, like t1 for the signal travelling from Tx to Rx1, the Tx and all Rxs should be synchronized. In in other words, the time instant when the signal leaves the Tx should be known to all Rx.

1.2.2 Time-difference-of-arrival (TDOA)

Much like the TOA, the TDOA methods use receivers that try to record the signal arrival time. However, in TDOA, the time instant of when the signal leaves the Tx is unknown. In order to do localization, the arrival time instant for each receiver is shared with all the other receivers. By setting a receiver as the reference (e.g., Rx1), the arrival time difference between the reference and the other receivers could be calculated. The arrival time difference between Rx1 and Rx3 is $\Delta t = t3 - t1$, or farther the distance difference $\Delta d = \Delta t * c$, where $t3$ and $t1$ are the arrival times for Rx1 and Rx3, respectively. With the distance difference Δd, a parabola can be plotted for Rx1 and Rx3, or it means the target should be on this parabola. With the same method, another parabola could be plotted for Rx1 and Rx2, and the intersection of the two parabolas identifies the location of the target. For the TDOA method, all receivers should be synchronized in order to calculate the arrival time difference.

1.2.3 Image processing

Matching is one of the main methods to localize in localization with image processing. The distance information is not directly estimated from this method. The idea of matching is usually used in image or video-based localization. The basic idea behind it is that a database of structured images of the indoor environment is constructed by using image matching and the bundle adjustment algorithm. Then each image's relative pose, like its position and orientation, is estimated and the semantic locations of images are tagged. The location of the user can then be determined by comparing a photo taken by the mobile phone to the database. This is done by combining quick image searching, matching, and the relative orientation.

1.2.4 Angle-of-arrival (AOA)

In AOA method, the receiver tries to identify the direction of the target transmitter. As shown in Fig. 1.3, the two receivers (Rx1 and Rx2) measure the direction of the Tx as θ_1 and θ_2 with a referenced direction. Therefore, the intersection of the two measured directions [10] is the location of the target.

1.3 Summary of contribution

In this dissertation, we will mainly focus on the position estimation in indoor localization with trilateration methods. Trilateration uses geometric distance to obtain the location of user. Therefore, distance estimation is among the key tasks in trilateration.

In RF signal-based indoor localization, as discussed earlier, TOA and TDOA are widely used to determine the distance-related information regarding the transmitter and receiver. However, several issues remain unsolved, like the synchronization, and non-line-of-sight (NLOS) problem, both of which are caused by the signal propagation nature of RF.

The first part of this dissertation focuses on resolving the issues of synchronization and NLOS problems. For the synchronization problem, we propose an algorithm to resolve the synchronization problem in the high speed multi-channel data acquisition unit in field-programmable gate array (FPGA). Armed with this algorithm, the sampling unit can sample four channels of its input signal at the speed of 3 Giga-samples per second (Gsps) while they are synchronized with each other. Also, we propose a synchronization-free positioning network architecture that eliminates the need for timing synchronization. For the NLOS problem, which greatly degrades positioning performance, we propose a semidefinite programing (SDP) process with a soft-minimal method and an NLOS link identification method with bias reduction to mitigate the NLOS error in TOA systems. For TDOA systems, NLOS mitigation is more difficult, since a reference should be fixed first. To overcome this problem, we propose a method to transform the TDOA architecture into a TOA architecture, and then form an SDP problem with new constraints. The proposed methods outperform the existing algorithms in both simulation and experiments.

To obtain the distance information, RF based techniques like the TOA and TDOA usually encounter several difficulties, as discussed above. We propose an image-tag based technique to obtain the distance as well as the angle information, which can completely avoid the most common issues in RF based localization and can increases the positioning accuracy. In the proposed method, after the segmentation of the tags from the image, information such as the tag ID, distance and angle with reference to the camera are retrieved through deep neural networks. The camera position is finally reliably and accurately estimated from such retrieved information.

1.4 Outline of the dissertation

Chapter 2 deals with the synchronization problem. In the first part of chapter 2, an algorithm is proposed and implemented in FPGA to achieve high speed, multi-channel, on-board synchronization for a data acquisition unit. In the second part, a synchronization free model is proposed. The architecture and its mathematical model are analyzed. The CRLB is derived and its performance is validated through the simulations. Although the simulated performance of the proposed model is little worse than the TOA and TDOA models, the synchronization free feature eliminates the complex synchronization requirements for a high accuracy localization system.

Chapter 3 analyzes the NLOS problem in both TOA and TDOA systems. As previous literature requires prior information about the NLOS to mitigate the NLOS error, the main focus in this chapter is to use convex optimization to mitigate the NLOS error without any prior information about NLOS. For the TOA system, at first, a revised SDP estimator with soft-minimum method is proposed. Then another two schemes with NLOS link identification and NLOS bias deduction are presented. For the TDOA system, as it's much more difficult to mitigate the NLOS error compared to that in the TOA system, the TDOA architecture is transformed into the TOA system and then is formed as a SDP problem with new constrains.

In RF signal-based localization with trilateration, the distance related information is affected by synchronization, multipath, and NLOS effects, which decrease the accuracy of localization. In chapter 4, the image-tag-based localization is proposed to estimate the localization with trilateration by estimating the distance and angle information. As the RF signal is not used, those difficulties dealing with synchronization and NLOS issues are totally avoided. The background of image based localization and its drawbacks are first introduced. Then a detailed flow chart of the image tag based localization system is proposed. Each part of the flow is analyzed in detail, with emphasis on tag segmentation, distance, and angle information extraction with deep neural networks, like the convolutional neural network (CNN). The performance is validated through experiments in the lab.

Chapter 2: Synchronization problem

2.1 Introduction

RF signal positioning using TOA or TDOA techniques generally requires timing synchronization within the anchors or between the target and anchors. If the desired position accuracy is high and coverage area is large, timing synchronization will be an extremely challenging issue.

For the TOA system, the transmitter and receiver pair should be synchronized. In the TDOA system, all receivers should be synchronized with each other. For low accuracy localization applications, synchronization is easy to achieve within micro second lever. But high accuracy positioning applications have a strong requirement for synchronization, and it's difficult to achieve. For example, ultra-wide band (UWB) signal based positioning requires nano-second level accuracy for synchronization.

In the first part of this chapter, we are trying to solve the TDOA-based synchronization problem in applications with a UWB signal. In the second part, a synchronization free scheme is discussed, which can completely remove the synchronization problem in the localization system by introducing the signal repeaters.

2.2 Synchronization in a high speed, multi-channe, data acquisition unit

In some localization applications with a TDOA architecture, several receivers are connected to the same data acquisition unit (DAU), which contains several analog-to-digital converters (ADCs) and can sample the signals simultaneously. In this case, the synchronization of TDOA is achieved through the DAU.

UWB impulse radio has been widely used in numerous indoor localization and navigation applications [14–16] due to its high time resolution. But UWB impulse system typically requires an extremely high sampling rate. For example, an one nanosecond width of pulse requires several gigahertz of capture rate. TDOA-based UWB localization system requires multiple synchronized channels [17]. However, due to minute differences that exist between any two oscillators, channel-to-channel synchronization is extremely difficult to achieve, since each channel typically uses its own oscillator [18]. Therefore, a design where all channels share the same clock source must be utilized in order to avoid small differences in clock frequency. In designing this high-speed multichannel data-acquisition unit (DAU), signal integrity (SI), routing delay control, and synchronization between channels must also be considered.

For the quandary of synchronization, although the manufacturers provide a method to synchronize channels by asserting a pulse, this method is costly and difficult to implement due to its strict timing requirements [19]. Other methods have been proposed to synchronize two-channel converters [20–22], but these methods are extremely difficult to implement with more than two channels.

Data transmission is also a difficult aspect of this design, as data can be generated at a rate of several GB/s. Additionally, in order to maintain signal reliability at the input and output of each transmission line, SI must be considered. Routing poses another challenge due to the strict routing delay requirements of a high-speed multichannel system and the large number of interconnects present in the design.

In this section, we introduce a synchronization method in FPGA in the high-speed, real-time, multi-channel data-acquisition unit. Our fabricated DAU has four synchronized channels, each of which is capable of sampling the input analog signal at a speed of up to 3 Gsps. The top-level block diagram for our system is shown in Fig 2.1, and the photograph of our newly developed DAU is shown in Fig. 2.2.

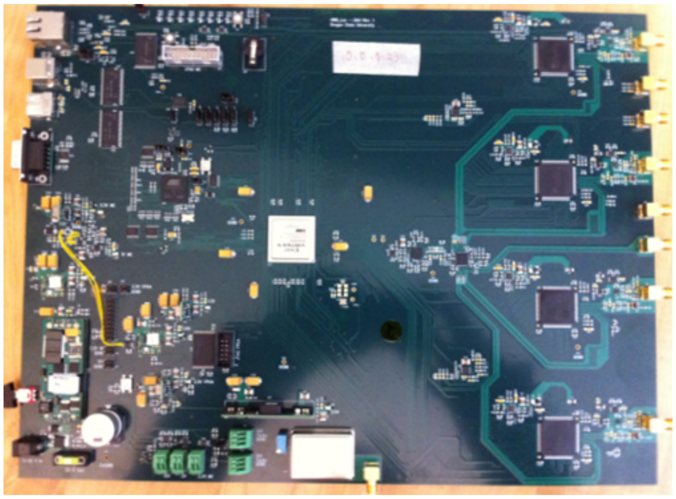


Figure 2.1: DAU fabricated.

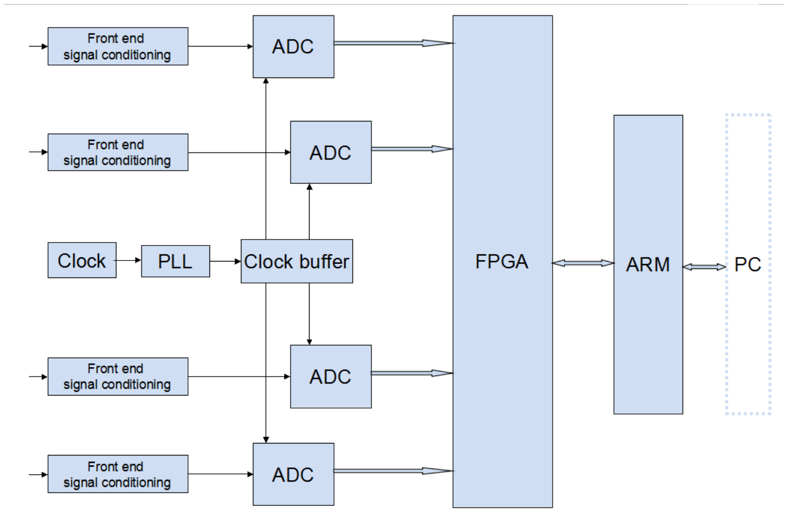


Figure 2.2: DAU diagram.

2.2.1 Motivation

2.2.1.1 Causes of challenges

For many applications, the channels of a multichannel data acquisition system should be synchronized so that the data in each channel are sampled at approximately the same time. For example, in TDOA-based localization, a small timing error may contribute to a large error in range estimation. It is therefore imperative to precisely synchronize each channel.

In our design, each ADC can sample the input analog signal at a speed of 3GHz with an 8-bit resolution. In order to guarantee that the high speed output data can be successfully accepted by the FPGA, the ADC concatenates four bytes together to form data with a width of 32 bits. If the double data rate (DDR) mode is used for the latching clock in the ADCs, its frequency can be reduced to approximately 375MHz by dividing the sampling clock frequency by four. Since each ADC will automatically concatenate four bytes of data (or four samples) together (such that the latching clock is 1/4 of the sampling clock), the phases of the four channels latching clock may be different. As Fig. 2.3 shows, the phase of two channels may have a difference of anywhere between zero and three clock cycles. This difference might change when the system is reset [19]. Therefore, it is not only critical to synchronize the input sampling clock to ensure identical sampling points, but also to synchronize output latching clocks. In addition, as the DAU is a high speed and mixed signal design, SI is an important concern. Because there are four channels

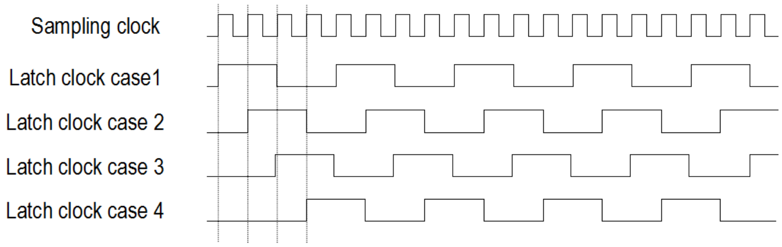


Figure 2.3: Four possible latching clock output states.

in this DAU (each with 66 interconnects), there is a total of 264 wires that need to be routed with approximately the same time delay between the ADCs and the FPGA. On top of these wires, there are 4 pairs of differential 1.5GHz clock wires between the clock buffer and ADCs.

2.2.1.2 Design challenges

There are many advantages to integrating high-speed multichannel capabilities in a single DAU. First, all channels share the same clock source and the clock frequency difference therefore does not exist. Second, it is easy to control and coordinate the execution of the four channels via the use of an onboard processor. Finally, each of the channels can be synchronized with one another, which guarantees proper sampling. However, designing the DAU poses significant technical challenges: 1) Synchronization of the high-speed, multichannel ADCs, 2) Storage of the large quantities of incoming data per second, 3) SI issues related to coupling of interconnects (especially those running at high frequencies), 4) Meeting design-specific routing delay constraints, 5) Keeping the system cool and properly dispersing heat away from the board components. The following sections will discuss the causes and related solutions to above challenges.

2.2.2 Synchronization of ADCs in a multichannel DAU

2.2.2.1 ADC synchronization in a high-speed, multichannel DAU

Fig. 2.4 shows the clock distribution path in our DAU. When the system is powered on, the original 10MHz clock is transformed into a 1.5GHz signal by the PLL. The clock buffer then distributes the 1.5GHz clock to four ADCs. Because our system's clock buffer can perfectly synchronize the four clock outputs with the input from the VCO, the four sampling clocks at the output of the clock buffer have the same phase. If the four sampling clocks are routed with approximately the same delay, the phases of the four sampling clocks at the input of four ADCs should still be aligned. Only when the four sampling clocks at the input of ADCs have already been synchronized, it is possible to synchronize the four latching clocks at the output of the ADCs.

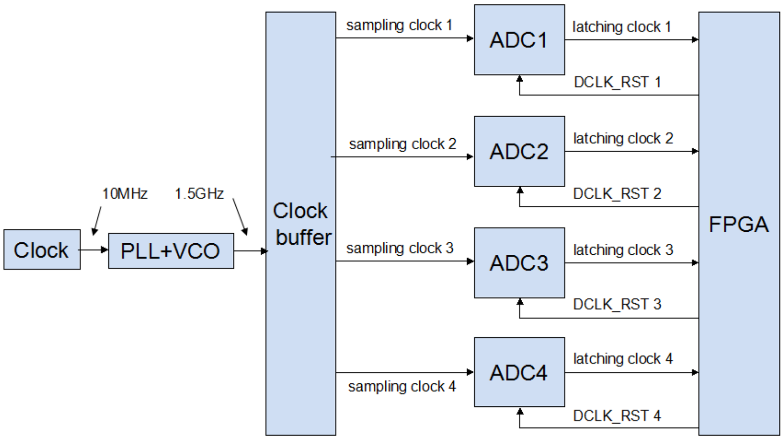


Figure 2.4: Clock distribution in our DAU.

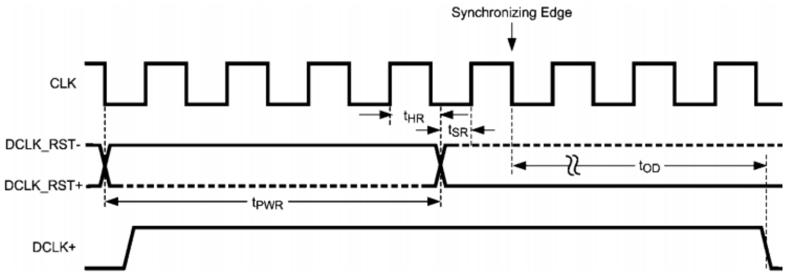


Figure 2.5: Timing requirements for DCLK_RST pulse to synchronize the multi-channels [19].

The problem of multi-chip synchronization has been considered by the manufacturer, but the requirements are often too strict to implement in practical circuit designs. Fig. 2.5 details the timing requirements for DCLK_RST. In this figure, CLK is the input sampling clock, t_{WR} is the minimal width of the pulse, t_{HR} is the hold time with respect to the sampling clock, t_{SR} is the setup time with respect to the sampling clock, and t_{DO} is the time delay before DCLK+. The latching clock from ADC, is synchronized to the sampling clock.

Synchronizing a high-speed, four-channel DAU using DCLK_RST poses two primary difficulties. First, when the sampling clock is 1.5GHz, a high speed circuit should be used to meet the setup and hold times with respect to 1.5GHz clock. Second, the routing delay constraints for four DCLK_RST pulses increase the routing difficulty.

The author of [20] offers three possibilities for multichannel synchronization. The first method works by "swallowing" clock pulses, and uses high speed logic to manipulate the sampling clock of one channel as long as the data clocks are not synchronized. The second method works by determining the offset between the data clocks and uses a barrel shifter to change the order of the input data for a single channel. The third method employs separate PLLs for each channel with the data clocks compared to a reference and used as feedback. In [21], the clock phases of two channels are compared and the phase offset is calculated. The data in memory will subsequently be shifted forward or backward according to the calculated phase offset. The author of [22] presents an idea to synchronize two clocks by resetting the converter through reset pin. Each reset operation will introduce a random clock phase. One of the chips is repeatedly reset until the clock signals of the two channels are aligned.

The ideas in [20] are not suitable for our case. And the method described in [21] is difficult to realize, as it requires keeping track of four distinct phase differences, which is a difficult and error-prone task. Furthermore, the position of the data in RAM should be updated in sample-by-sample according to the phase offset. Adding a single additional channel can greatly increase the complexity of these methods. While the concept in [22] seems promising, it is impossible to implement without a built-in reset function (a function that the ADC083000 does not provide) and can become time consuming and unreliable as the number of channels increases.

Unlike the methods mentioned above, our method requires only two DCMs and a small amount of FPGA logic to perform synchronization. According to the timing requirements for DCLK_RST, only setup and hold time are difficult to

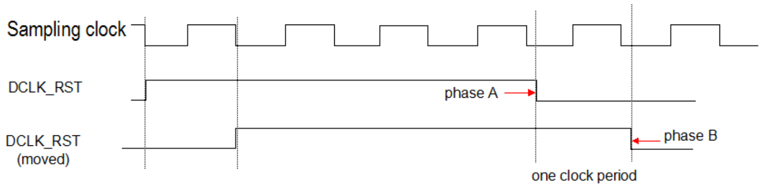


Figure 2.6: Move the pulse of DCLK_RST from phase A to phase B, step by step.

meet, while the others pose no technical challenge. The basic idea is to generate a DCLK_RST pulse and subsequently move the pulse forward in time step-by-step, as shown in Fig. 2.6. When the pulse is moved across one period of the sampling clock, there must be some pulse with a phase that can meet the setup and hold timing, assuming the step size is small enough.

Based on this idea, the basic procedure to synchronize the multichannel latching clocks is as follows: 1) Set the latching clock of the first channel as the reference clock and keep it unchanged. Assert a DCLK_RST pulse to the second channel; 2) Calculate the phase difference between channels one and two and judge whether they have already been synchronized. If yes, go to step 4. If not, proceed to the next step; 3) Move DCLK_RST for channel two a step forward. Repeat step 2 until the two channels are synchronized; 4) Keep first channel unchanged and assert a DCLK_RST pulse to the third channel. Repeat step two until the third and the first channel are synchronized; 5) Repeat the procedure for the remaining channels until all of them are synchronized; The following sections discuss the above procedures in detail.

2.2.2.2 Pulse shifting and phase difference calculation

For the ADC083000, when the input sampling clock is 1.5GHz, the latching clock will have a frequency of 375MHz in DDR mode. The DCLK_RST pulse can be easily generated on the FPGA with the latching clock. However, the phase of the DCLK_RST pulse cannot be shifted directly in the FPGA. Here, we introduce a method of indirectly moving the pulse. The Advanced DCM in each FPGA is able to shift the phase of the input clock backward or forward with a resolution of 1/256 clock periods, assuming the input clock frequency is greater than 100MHz. Furthermore, the phase shifting can be done continuously, and the maximal range for a phase shift is 10ns when in POSITIVE MODE [23].

As shown in Fig. 2.7, the DCLK_RST pulse is always generated by the shifted clock from a DCM. Therefore, the phase of the pulse is shifted when the phase of the clock is shifted. If the 375MHz latching clock is used, the resolution of each step will be about 10ps. Therefore, each sampling clock period includes about 60 steps when its frequency is 1.5GHz. That means that, after 60 phase shifts, there must be some pulses whose falling edge can meet the setup and hold time requirements for DCLK_RST. That's because the required falling edge is somewhere between two adjacent rising edges of the sampling clock.

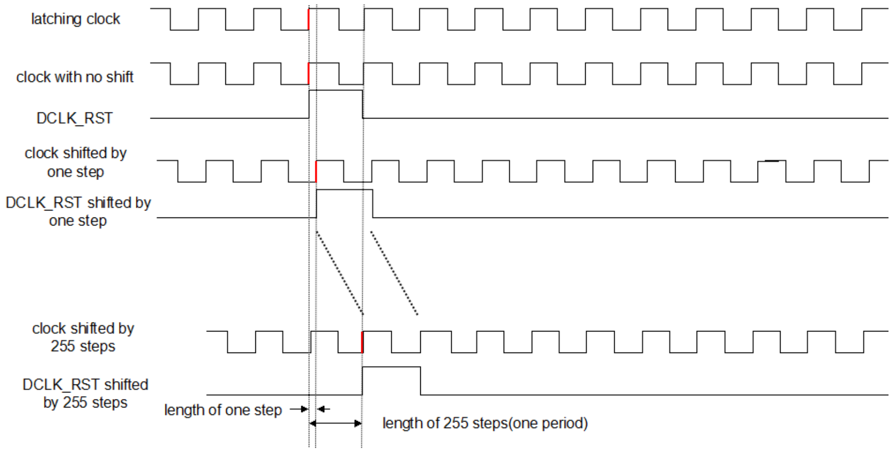


Figure 2.7: Method to shift the phase of DCLK_RST.

Fig. 2.8 shows a way to calculate the phase difference between two channels via the use of a second DCM. Without lossing generality, assume that there are N channels that need to be synchronized. Set the first channel's latching clock to be the reference clock. The basic procedure for calculating the phase difference between two channels is as follows: 1) The phase of the latching clock for channel one will be shifted forward by one step using the Advanced DCM and transmitted

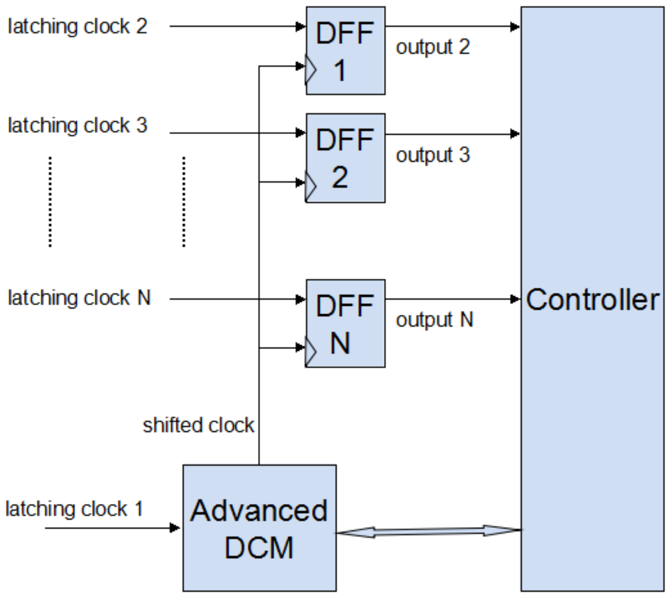


Figure 2.8: Method of calculating two channels' phase difference in FPGA.

to each of the D flip-flops (DFFs) as clock input; 2) Each DFF samples the input. The output of each DFF represents a high or low state of the latching clock; 3) For each DCM shift, the controller counts one time when output of the DFF 1 is high and continues unchanged when the output is low; 4) Repeat above procedures 127 times (since only half the period of latching clock, which contains 128 steps, must be swept through) and check the sum of the counter. If the counter is greater than a threshold value, then the two channels have been synchronized. If not, a shifted DCLK_RST will be asserted and the above procedure will restart;

Fig. 2.9 shows how a single DFF samples the latching clock using the shifted clock. If the phases of channel 1 and 2 are perfectly aligned, the output of DFF 1 will be always high when the shifted clock is shifted from 1 to 128 steps. In that case, the counter value will be 128. On the other hand, if the two channels are not aligned, the counter will be some value below 128. Using this value, we can judge whether the two clocks have been aligned.

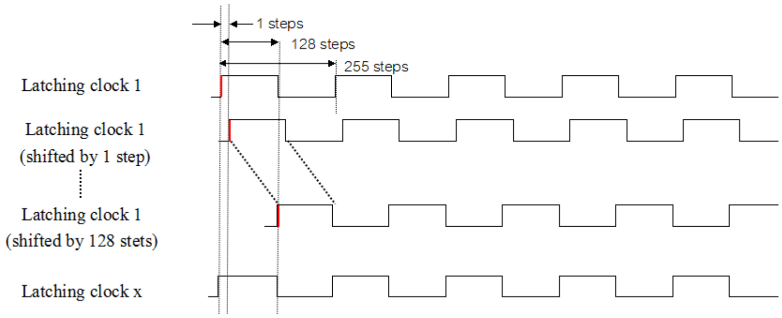


Figure 2.9: Description of how the shifted clock samples the latching clock 2.

2.2.2.3 Writing synchronized data into memory

At this point, the N channels have been synchronized. In an ideal case, these channels should have identical phases when processed by the FPGA. However, in reality, this is impossible to accomplish due to the small differences in routing delay between each channel. This poses another challenge to synchronization when writing data into memories within the FPGA. As a result, writing the output data into memory poses another difficulty to the system.

In the FPGA, each channel has its own memory to temporally store the sampled data. In order to save the data at the same time, a general write-enable signal for each memory is necessary. As shown in Fig. 2.10, W_EN is the write-enable signal for memory and latching clocks from 1 to N have small phase difference due to the different routing delays in the PCB. The first and last rising edge of each latching clock forms the clock rising edge distribution area. Each channel's memory is composed of block memory in the FPGA to temporally store the sampled data. In order to store the data concurrently, a W_EN is asserted to all memories at roughly the same time. As it's possible that the rising edge of W_EN may fall in the rising edge distribution area of a latching clock (see Fig. 9), the rising edge of W_EN may be asserted after the rising edge of clock N but before all the others. In this scenario, the data of the Nth channel will not be stored during this clock period but the other channels' data will.

An effective way to avoid this problem is to move the rising edge of W_EN to a time that is always outside of the latching clock's rising edge distribution area. To accomplish this, we AND each of the latching clocks to get a new signal. For simplicity, we will refer to this signal as Clock_AND. We subsequently use the falling edge of the new clock to resample W_EN, thereby forcing the rising edge of the W_EN to a position shortly after the falling edge of Clock_AND, as shown in Fig. 2.10. If the new write enable signal is applied to all memories, data from each clock period will be stored into the memory simultaneously.

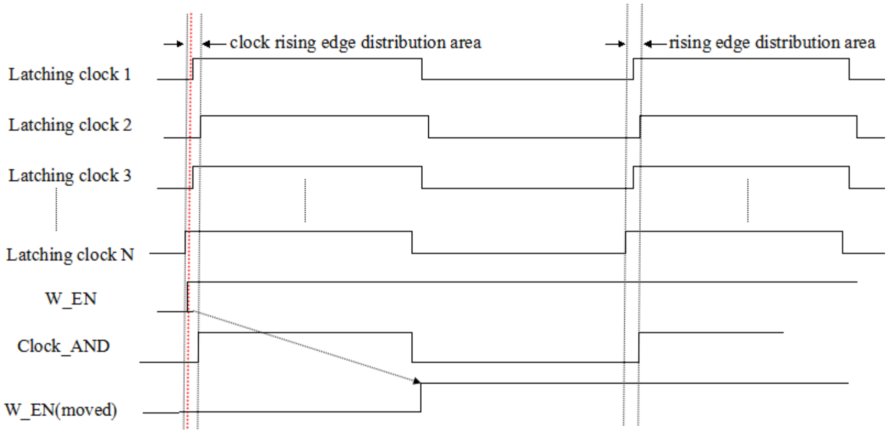


Figure 2.10: A proper way to write the data into memory (assume that the rising edge of latching clock is used and the HIGH state of W_EN is active).

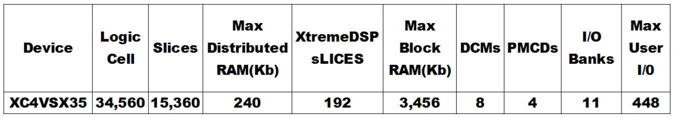
2.2.3 Implementation

2.2.3.1 Design implementation

As shown in Fig. 2.2, in this DAU, the four single end analog input signals are fed into four front end signal conditioning circuits, where the analog signal is modified according to the requests and then transmitted to ADC. The high speed data generated by the four ADCs are then transmitted to FPGA and stored temporally. The data can then be transferred to a computer through the Ethernet or USB port.

The system combines a Xilinx Vertex-4 xc4vsx35 FPGA with an AT91SA9G20

ARM processor to maximize the system's performance. Taking advantage of the ARM's management ability, we allow the processor to drive the system and guarantee that the DAU performs properly.. The FPGA in this system is used to perform high speed data transmission, data processing, temporal data storage, and synchronization of the four channels. These requirements can be guaranteed by the rich resources in this FPGA, as shown in Fig. 2.11. The abundant amount of Block RAM in the FPGA enables it to act as a large, high speed data cache. Furthermore, the 8 DCMs inside the FPGA are important for clock management and channel synchronization while the large amount of DSP slices are left for high speed data processing.



Device	Logic Cell	Slices	Max Distributed RAM(Kb)	XtremeDSP sLICES	Max Block RAM(Kb)	DCMs	PMCDs	I/O Banks	Max User I/0
XC4VSX35	34,560	15,360	240	192	3,456	8	4	11	448

Figure 2.11: Resource of Virtex-4 xc4vsx35 FPGA.

To sample the input analog signal, our system utilizes the ADC083000 chip from Texas Instruments (TI). Each of the ADCs works in double edge sampling mode and can sample at a rate of up to 3 Gsps (the input sampling clock operates at 1.5 GHz). The PLL (we use an LM2531 in our design) and clock buffer are used together to generate the required clock frequency and distribute it to the four ADCs. The LM2531 PLL is a high performance frequency synthesizer integrated with VCO. For the LM2531, there are two frequency bands available–low band and high band. In low band, the output frequency can be set anywhere from 749.5MHz to 755MHz. In high band, it can be set from 1499MHz to 1510MHz. To distribute the clock, we use an LMK01000 clock buffer. This chip transmits its input clock to all four ADCs with the same phases between outputs (i.e., all ADCs see the same phase at their sampling clock input).

The single-ended analog inputs are modified by the front end signal conditioning circuits in order to meet the needs of ADC, such as differential input and voltage range. Two alternative analog pre-processing circuits are available and can be selected by a jumper. One of the circuits has a filter to limit the frequency to under 1.5GHz and the other allows high frequency, band-pass signals to pass to the ADC.

The circuit board was designed using the Mentor Graphics PADS software tool. By physically laying out the components on the board according to the rules stated, the maximum routing delay difference between the four sampling clocks from the clock buffer to ADCs is less than 0.013ns and the routing delay difference between the four latching clocks is approximately 0.124ns. The maximum difference between each latching clock and its related data wires are ($-327 \sim 80$ps), ($-239 \sim -54$ps), ($-342 \sim 8$ps), and ($-434 \sim -16$ps) for ADCs one to four, respectively. All routing delay data is given by the PADS Layout software.

2.2.3.2 Implementation of synchronization and data transmission in FPGA

Synchronization of the four channels is implemented on the FPGA using three state machines, as shown in Fig. 2.12. State machine one guarantees the channels are synchronized, one by one. The second state machine conducts DCLK_RST, shifting with a maximum of 60 shifting steps for each channel. After each shifting procedure in state machine two, state machine three will shift the latching clock for 128 steps and calculate the phase difference between the channels. Using ChipScope [24]4, a program packaged in the Xilinx ISE software suite, we were able to determine that the tasks in state machine three take approximately 96 micro-second to perform. Therefore, the maximum amount of time used to finish synchronizing four channels is approximately 96×60(shifting steps)$\times$(4-1)(channels), which equals 18ms.

Since each ADC provides data at a rate of 3GB/s, the ADC bank will produce data at a total rate of 12GB/s [19]. In order to properly transmit and store the data, we employ a parallel method shown in Fig. 2.13. In the FPGA, the 32-bit width differential inputs from each ADC are first fed through a differential input buffer (IBUFDS) to acquire single-ended data. Input double-data-rate registers

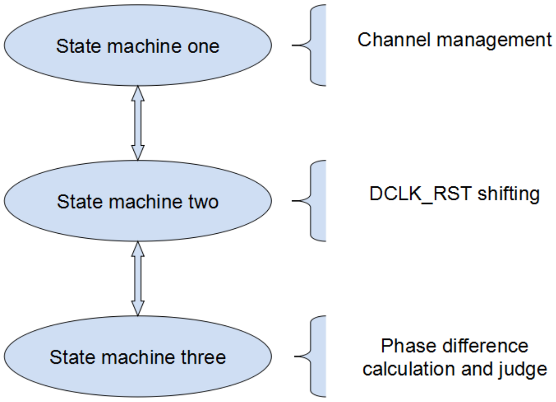


Figure 2.12: The three level state machines used to implement synchronization in FPGA.

(IDDRs) are then used to transfer the data from DDR mode to single-data-rate (SDR) mode. The logic after IDDR packs the two 32-bit width output from IDDR to form a 64-bit width data. The 64-bit width data is then written into Block RAM (BRAM) for temporary storage. Each channel runs independently and in parallel. The dual port BRAM in the FPGA is an ideal candidate with which to store high speed data, as it can work at a speed of up to 400MHz and store up to 3,456KB (432KB) of data.

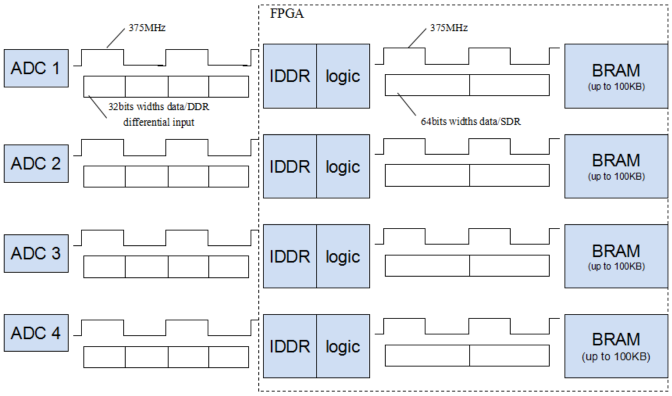


Figure 2.13: Data transmission and processing.

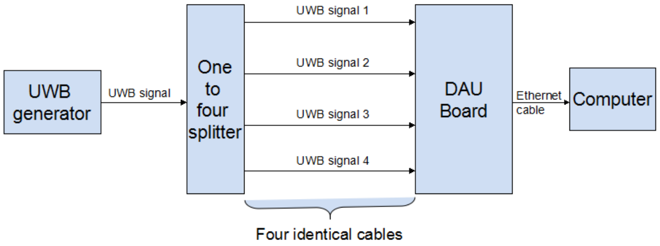


Figure 2.14: System level implementation.

2.2.4 Experiment and results

Fig. 2.14 shows a system level experiment using UWB pulses. In the experiment, the signal from an UWB signal generator is fed into a one-to-four splitter. The four identical UWB signals from each splitter are transmitted to the DAU via four identical cables. And in the DAU, the signal is sampled, temporally stored and finally transmitted to computer via Ethernet cable. The UWB pulse has a band width of about 0.7GHz in frequency domain and has a width of approximately one nanosecond (ns) in time domain. The input sampling clock is set to 1.5GHz and therefor the ADCs will work at 3 Gsps in DDR mode, which guarantees the Nyquist Sampling Theorem in sampling the UWB pulse. Since the width of each pulse is about 1ns, three samples can be obtained for each UWB pulse. If the four channels are well synchronized, the sampled signal should have identical shape.

Fig. 2.15 shows the result of the above experiment, where the original UWB signal has a total width of about 1ns, as shown on the left. The plot on the right shows the amplitude of the sampled data in each channel. From here, its easy to see that the four channels have already been synchronized.

2.3 Synchronization free model

2.3.1 Introduction

In section 2.2, the multichannel synchronization problem could be achieved through the on board FPGA, as all channels are connected on the same board. If each

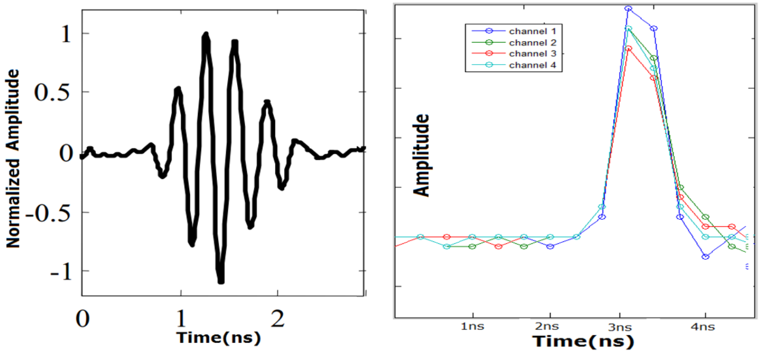


Figure 2.15: UWB signal (left) and the signal sampled by four channels (right).

channel is distributed in a different location, the synchronization of these channels will become really difficult [32]. To cover a large indoor area that requires a number of anchors, anchor-synchronization becomes extremely difficult.

Common synchronization techniques include reference broadcast synchronization (RBS) [34], pair-wise synchronization [35], master-slave synchronization [36], and methods in some special applications [37–39]. Time synchronization with above methods is generally obtained based on a sequence of exchange of beacon messages. The message that carries timing information often suffers from various delay and uncertainties before it reaches other nodes. In [33] the timing uncertainties are categorized into four components:

- Send time: The time used to construct a beacon message at the sender;
- Access time: The delay at the medium access control (MAC) layer before actual transmission;
- Propagation time: The time it takes a message to travel from the sender to the receiver;
- Receive time: Time needed for a receiver to receive and decode the message, and report it to upper layers.

The solutions in [40, 41] are able to resolve some of these issues, and in [42] a two-signal sensing method with acoustic signals is proposed to resolve these issues.

In practical systems, issues like clock skew and drifting must also be considered, which means that timing synchronization is not a one-time effort, but needs to be done continuously.

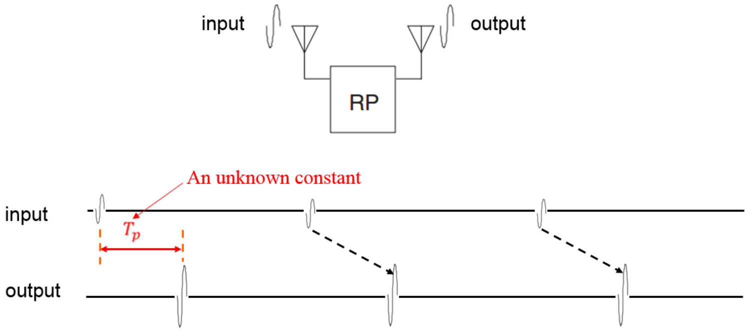


Figure 2.16: Signal repeater.

We introduce signal repeaters into the network to construct a new timing model. An example of a signal repeater is shown in Fig. 2.16. It receives the signal from the input and amplifies and transmits it with some unknown constant processing time. The basic idea is to introduce a repeater whose location is fixed to work with its nearby anchors; for networks with a large number of anchors, more repeaters can be deployed but each works with only a subset of its neighboring anchors. The target exchanges timing information with only a subset of anchors and the repeater that it can reach. The key advantage is that *all the anchors and the repeaters work independently*.

Compared with existing work, our contributions include: (1) The problems associated with timing synchronization as discussed above are completely resolved by the proposed new model; (2) The system structure and network implementation are greatly simplified; (3) A mathematical model of this scheme is developed and its Cramér-Rao lower bound (CRLB) is analyzed, together with simulation results to justify that the mean squared error (MSE) of the proposed scheme with the nonlinear least-squares (NLLS) positioning method can approach the CRLB; and (4) The model can be flexibly extended to cover larger areas simply by introducing more signal repeaters into the network, since all anchors and repeaters work independently with the proposed scheme. However, with the same network configuration and other system parameters, the positioning performance of the proposed scheme is not as good as that of existing schemes assuming *perfect timing-synchronization*. In any case, the performance loss is small, as will be seen from the numerical results.

2.3.2 System description

For clarity, a simple network with one anchor, which is assumed to be a receiver, one repeater, and one transmitter, as shown in Fig. 2.17, is used to describe how the proposed scheme works. There are many ways to acquire the timing information between any transmitter-receiver/repeater pair, and here we assume the transmitter sends pulses (or any short burst of radio frequency signals) for timing information. The transmitted pulse arrives at the receiver via two paths: a direct path (Path 1) and a path via the repeater (Path 2). The time instants when the transmitted positioning signal arrives at the repeater and receiver as well, as the time instant when it leaves the repeater, are shown in Fig. 2.18: T_f is the repetition period of the transmitter pulses, t_1 and t_2 are, respectively, the times it takes the transmitted signal to reach the receiver and the repeater, t_3 is the time it takes the signal from the repeater to the receiver, and T_p, a constant, is the time the repeater needs to process its received signal. The time difference between the

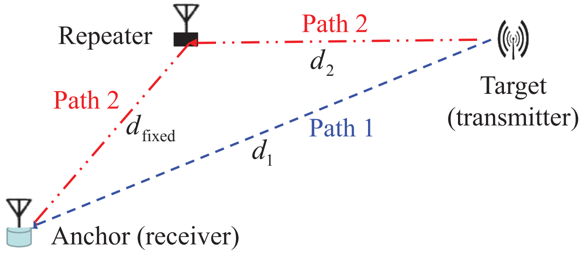


Figure 2.17: A simple network with one receiver, one repeater, and one transmitter.

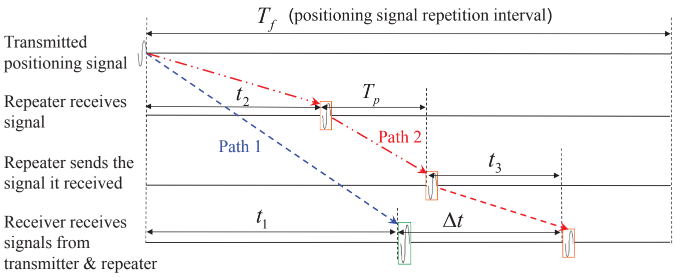


Figure 2.18: Time instants when the transmitted signal arrives and leaves.

two paths is expressed as

$$\Delta t = t_2 + t_3 + T_p - t_1. \tag{2.1}$$

To avoid overlaps between the signals from path 1 and path 2, it is required that $\Delta t \leq T_f/2$.

The distance difference between the two paths is written as

$$\Delta d = d_{\text{fixed}} + d_2 - d_1 = (\Delta t - T_p)c, \tag{2.2}$$

where c is the speed of light and d_{fixed} is a known constant. The difference between the transmitter-position-dependent distances, with d_1 and d_2, is expressed as

$$\Delta d' = d_2 - d_1 = \Delta d - d_{\text{fixed}}. \tag{2.3}$$

Therefore, for each receiver-repeater pair, one set of range measurements d_1 and d_2 could generate one hyperbola where the transmitter might be located. This is illustrated in Fig. 2.19.

The analysis above shows that with the proposed scheme, for 2-dimensional (2D) positioning, at least two receivers are required; for 3-dimensional (3D) positioning, at least three receivers are required. A simple example in 2D with four receivers is shown in Fig. 2.20. The major advantage is that the receivers and the repeater work independently with their own clocks; that is, no time-synchronization

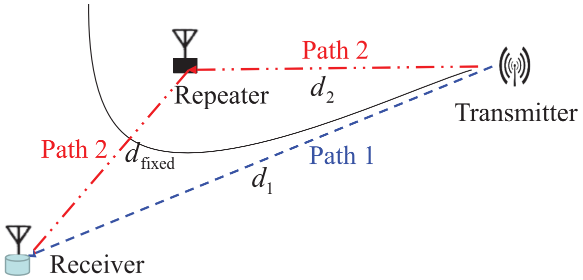


Figure 2.19: Time-difference model for the proposed scheme.

is required among the receivers and the repeater. Also, the proposed model is very easy to implement, and it does not require any extra communications among the transmitter, repeater, and receiver to exchange messages like the methods described in [40–42]. Since only a local clock is used, there are no clock skew and clock drifting problems.

A potential issue of the proposed model is that, although the signal from the repeater will always arrive at the receiver after the signal directly sent by the transmitter arrives, there is a possibility that the signal sent by the transmitter may not be distinguishable from the signal sent by the repeater, if the transmitter sends signals continuously. A simple way to resolve this issue is to increase the transmission interval of the transmitted signal. For example, the transmission interval T_f is set to be at least twice as large as the maximum value of Δt as shown in Fig. 2.18. In this way, the transmitter can send signals continuously while the receiver only needs to find any pair of signals that are close to each other in time from a block of sampled signals. Therefore, unlike existing schemes, with the proposed model, positioning can be continuously without interruptions. This greatly simplifies the network implementation.

2.3.3 Analysis

In this section, we build the mathematic model for 2D localization and analyze the CRLB. Extending the analysis to 3D localization is straightforward.

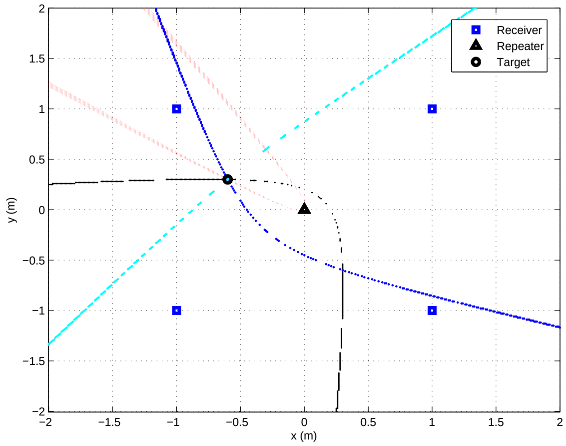


Figure 2.20: An example for 2D localization with the proposed scheme assuming four receivers.

2.3.3.1 Mathematical model

The network, as shown in Fig. 2.21, consists of M receivers at known locations $(x_i, y_i)^T, i = 1, 2, \cdots, M$, and one repeater at (x_0, y_0), which is assumed to be located somewhere in the middle of the receivers. The processing time at the repeater T_p is a constant, but unknown. The goal is to estimate the unknown source location $\theta = (x, y)^T$ without requiring time-synchronization among the receivers and the repeater.

For the receiver i and repeater pair, the distance difference between the two paths, d_{i1} and d_{i2}-repeater-d_{i3} as shown in Fig. 2.21, is expressed as

$$\begin{aligned} \hat{d}_i &= (d_{i2} + n_{i2}) + d_{i3} + d_p - (d_{i1} + n_{i1}) \\ &= (d_{i2} + d_{i3} + d_p - d_{i1}) + (n_{i2} - n_{i1}) \\ &= d_i + n_i, \quad i = 1, 2, 3, \cdots, M, \end{aligned} \tag{2.4}$$

where n_{i1} and n_{i2} are, respectively, the ith receiver's distance measurement errors for paths 1 and 2, which are assumed to be independent Gaussian variables. Also,

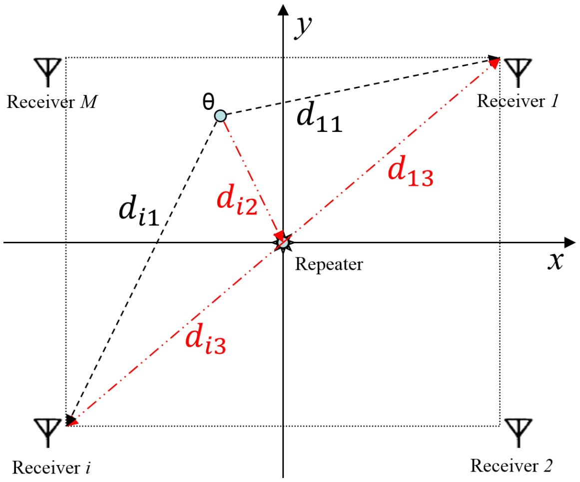


Figure 2.21: The network for the analysis.

d_{i1}, d_{i2}, and d_{i3} are expressed as

$$d_{i1} = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \tag{2.5a}$$

$$d_{i2} = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \tag{2.5b}$$

$$d_{i3} = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}. \tag{2.5c}$$

For one set of observations from M receivers, the observation vector can be written as

$$\begin{aligned}\hat{\boldsymbol{d}} &= \left[\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_M\right]^T \\ &= [d_1, d_2, \cdots, d_M]^T + [n_1, n_2, \cdots, n_M]^T \\ &= \boldsymbol{d} + \boldsymbol{n}, \end{aligned} \tag{2.6}$$

where $[\cdot]^T$ denotes transpose and $\hat{d}_i, d_i, n_i, i = 1, \cdots, M$, are given in Eq. (2.4). Note that in Eqs. (2.4) and (2.6), $d_{12} = d_{22} = \cdots = d_{M2}$.

Let $D_i = (\hat{d}_i - d_i), i = 1, \cdots, M$, which are zero-mean Gaussian random variables, and $\sigma_i^2 = E[D_i^2] = \sigma_{i1}^2 + \sigma_{i2}^2, i = 1, \cdots, M$, where $E[\cdot]$ denotes expectation. Also, let $\boldsymbol{D} = [D_1, D_2, \cdots, D_M]^T$ whose covariance matrix is expressed as $\boldsymbol{C} = E[\boldsymbol{D}\boldsymbol{D}^T]$. For each set of range observations, $\boldsymbol{C}$ is written as

$$\boldsymbol{C} = \begin{bmatrix} \sigma_{12}^2 + \sigma_{11}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{22}^2 + \sigma_{21}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{M2}^2 + \sigma_{M1}^2 \end{bmatrix}. \tag{2.7}$$

2.3.3.2 Cramér-Rao lower bound

The probability density function (pdf) of $\hat{\boldsymbol{d}}$ given a set of range observations is expressed as

$$p(\hat{\boldsymbol{d}}; \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{C}|}} \exp\left(-\frac{1}{2}\left[\hat{\boldsymbol{d}} - \boldsymbol{d}\right]^T \boldsymbol{C}^{-1} \left[\hat{\boldsymbol{d}} - \boldsymbol{d}\right]\right), \tag{2.8}$$

where $\boldsymbol{C}$ and $\boldsymbol{d}$ depend on $\boldsymbol{\theta}$ and $|\boldsymbol{C}|$ is the determinant of $\boldsymbol{C}$. The (i, j)-th element of the Fisher information matrix (FIM) is expressed as [43]

$$[\boldsymbol{I}(\boldsymbol{\theta})]_{ij} = \left[\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \theta_i}\right]^T \boldsymbol{C}^{-1}(\boldsymbol{\theta}) \left[\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \theta_j}\right] + \frac{1}{2}\mathrm{tr}\left[\boldsymbol{C}^{-1}(\boldsymbol{\theta}) \frac{\partial \boldsymbol{C}(\boldsymbol{\theta})}{\partial \theta_i} \boldsymbol{C}^{-1}(\boldsymbol{\theta}) \frac{\partial \boldsymbol{C}(\boldsymbol{\theta})}{\partial \theta_j}\right], \tag{2.9}$$

where tr[·] denotes trace.

For the special case of independent and identically distributed (i.i.d.) range errors, $\boldsymbol{C}$ does not depend on $\boldsymbol{\theta}$, and Eq. (2.9) simplifies to

$$\left[\boldsymbol{I}(\boldsymbol{\theta})\right]_{ij} = \left[\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \theta_i}\right]^T \boldsymbol{C}^{-1} \left[\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \theta_j}\right]. \tag{2.10}$$

In this case, for 2D positioning, $\boldsymbol{\theta} = (x, y)^T$, and the FIM is written as

$$\text{FIM} = \begin{bmatrix} [\boldsymbol{I}(\boldsymbol{\theta})]_{11} & [\boldsymbol{I}(\boldsymbol{\theta})]_{12} \\ [\boldsymbol{I}(\boldsymbol{\theta})]_{21} & [\boldsymbol{I}(\boldsymbol{\theta})]_{22} \end{bmatrix}. \tag{2.11}$$

From Eq. (2.5) it is straightforward to obtain the following relationships:

$$\partial d_i/\partial x = (x - x_0)/d_{12} - (x - x_i)/d_{i1}, \tag{2.12a}$$

$$\partial d_i/\partial y = (y - y_0)/d_{12} - (y - y_i)/d_{i1}. \tag{2.12b}$$

Therefore,

$$\frac{\partial \boldsymbol{d}}{\partial x} = \begin{bmatrix} \frac{x-x_0}{d_{12}} - \frac{x-x_1}{d_{11}} \\ \frac{x-x_0}{d_{12}} - \frac{x-x_2}{d_{21}} \\ \vdots \\ \frac{x-x_0}{d_{12}} - \frac{x-x_M}{d_{M1}} \end{bmatrix}; \quad \frac{\partial \boldsymbol{d}}{\partial y} = \begin{bmatrix} \frac{y-y_0}{d_{12}} - \frac{y-y_1}{d_{11}} \\ \frac{y-y_0}{d_{12}} - \frac{y-y_2}{d_{21}} \\ \vdots \\ \frac{y-y_0}{d_{12}} - \frac{y-y_M}{d_{M1}} \end{bmatrix}. \tag{2.13}$$

The elements of the FIM in Eq. (2.11) are expressed as

$$[\boldsymbol{I}(\boldsymbol{\theta})]_{11} = [\partial \boldsymbol{d}/\partial x]^T \boldsymbol{C}^{-1} [\partial \boldsymbol{d}/\partial x], \tag{2.14a}$$

$$[\boldsymbol{I}(\boldsymbol{\theta})]_{22} = [\partial \boldsymbol{d}/\partial y]^T \boldsymbol{C}^{-1} [\partial \boldsymbol{d}/\partial y], \tag{2.14b}$$

$$[\boldsymbol{I}(\boldsymbol{\theta})]_{21} = [\boldsymbol{I}(\boldsymbol{\theta})]_{12} = [\partial \boldsymbol{d}/\partial x]^T \boldsymbol{C}^{-1} [\partial \boldsymbol{d}/\partial y]. \tag{2.14c}$$

The CRLBs are calculated as

$$\text{CRLB}(x) = [\boldsymbol{I}^{-1}(\boldsymbol{\theta})]_{11} \tag{2.15a}$$

$$\text{CRLB}(y) = [\boldsymbol{I}^{-1}(\boldsymbol{\theta})]_{22}. \tag{2.15b}$$

2.3.4 Numerical results

It would be informative to compare the CRLB of the proposed scheme, which does not require time-synchronization of the anchors, with those of the traditional TOA and TDOA schemes assuming *perfectly* synchronized anchors. For this, consider the specific setting in Fig. 2.22, where a repeater is placed in the middle of the four anchors; for TOA and TDOA schemes, the repeater is absent. The absolute dimension of the space covered does not matter, because we are comparing the relative CRLBs of various schemes. The range estimation error variance is chosen to be $\sigma^2 = 0.01$.

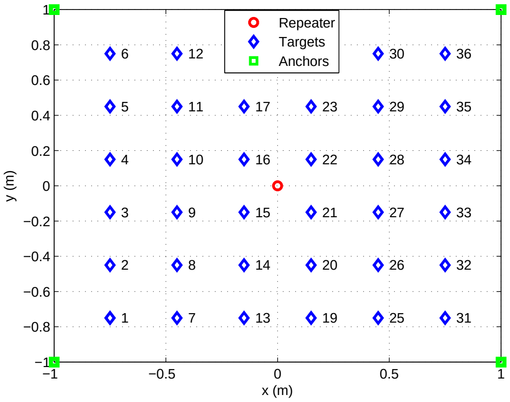


Figure 2.22: Setting for CRLB calculation and positioning MSE simulation.

The CRLBs are shown in Figs. 2.23 and 2.24, for the x- and y-dimensions, respectively. It is observed that, compared with TOA and TDOA, the CRLB of the proposed scheme is higher. Therefore, the synchronization-free property comes at the price of some localization performance loss.

Since most practical estimators would not reach the CRLB, it is also informative to see how close is the positioning MSE to the CRLB. The NLLS estimator is chosen here for MSE simulation since, given the range error model adopted, it performs the same as the maximum likelihood estimator. For each observation, Eq. (2.6) can be formed. The least squares (LS) estimate for $\boldsymbol{\theta}$ is expressed as

$$\hat{\boldsymbol{\theta}}_{LS} = \arg\max_{\boldsymbol{\theta}} \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta})\right]^T \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta})\right]. \tag{2.16}$$

Since $\boldsymbol{d}(\boldsymbol{\theta})$ is a non-linear function of $\boldsymbol{\theta}$, we linearize it using Taylor series for an

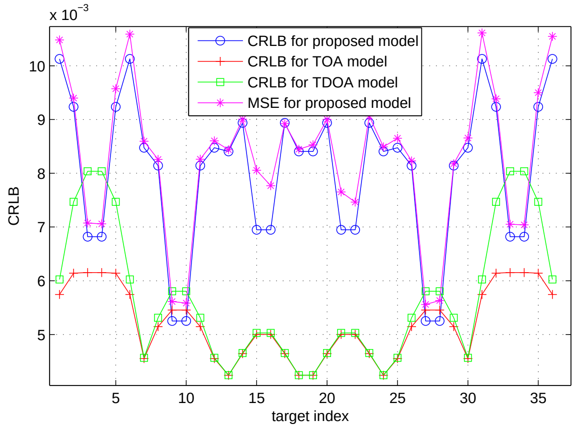


Figure 2.23: CRLB and MSE for the x-dimension.

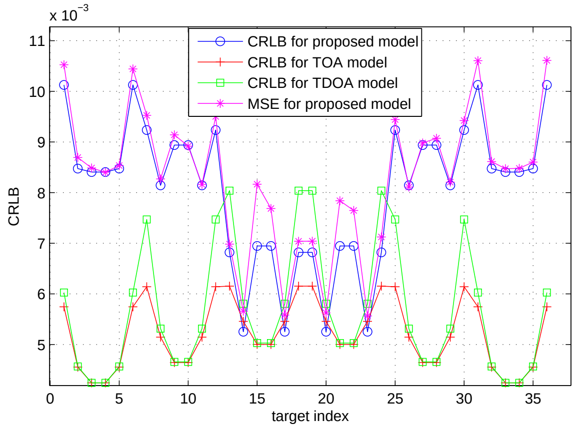


Figure 2.24: CRLB and MSE for the y-dimension.

initial value $\boldsymbol{\theta}_{in} = [x_{in}, y_{in}]^T$:

$$\boldsymbol{d}(\boldsymbol{\theta}) \approx \left[\boldsymbol{d}(\boldsymbol{\theta}_{in}) + \left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{in}} (\boldsymbol{\theta} - \boldsymbol{\theta}_{in}) \right]. \tag{2.17}$$

Substituting Eq. (2.17) into Eq. (2.16), we have

$$\hat{\boldsymbol{\theta}}_{LS} \approx \arg\max_{\boldsymbol{\theta}} \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta}_{in}) - \left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{in}} (\boldsymbol{\theta} - \boldsymbol{\theta}_{in}) \right]^T \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta}_{in}) - \left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{in}} (\boldsymbol{\theta} - \boldsymbol{\theta}_{in}) \right], \tag{2.18}$$

where

$$\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{x-x_0}{d_{12}} - \frac{x-x_1}{d_{11}} & \frac{y-y_0}{d_{12}} - \frac{y-y_1}{d_{11}} \\ \vdots & \vdots \\ \frac{x-x_0}{d_{12}} - \frac{x-x_1}{d_{i1}} & \frac{y-y_0}{d_{12}} - \frac{y-y_1}{d_{i1}} \\ \vdots & \vdots \\ \frac{x-x_0}{d_{12}} - \frac{x-x_1}{d_{M1}} & \frac{y-y_0}{d_{12}} - \frac{y-y_1}{d_{M1}} \end{bmatrix}. \tag{2.19}$$

Let

$$\boldsymbol{X} = \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta}_{in}) + \left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{in}} \boldsymbol{\theta}_{in} \right], \tag{2.20a}$$

$$\boldsymbol{H} = \left[\left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{in}} \right]. \tag{2.20b}$$

Eq. (2.18) allows us to obtain the LS solution:

$$\hat{\boldsymbol{\theta}}_{LS} = \left[\boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{H} \right]^{-1} \boldsymbol{H}^T \boldsymbol{C}^{-1} \boldsymbol{X}, \tag{2.21}$$

where $\boldsymbol{C}$ is given in Eq. (2.7).

The NLLS estimator can be implemented iteratively as

$$\begin{aligned}\boldsymbol{\theta}_{k+1} =& \boldsymbol{\theta}_k + \left[\left[\left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}\right]^T \boldsymbol{C}^{-1} \left[\left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}\right]\right]^{-1} \\ & \left[\left[\left.\frac{\partial \boldsymbol{d}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}\right]^T \boldsymbol{C}^{-1} \left[\hat{\boldsymbol{d}} - \boldsymbol{d}(\boldsymbol{\theta}_k)\right]\right]. \end{aligned} \tag{2.22}$$

The setting in Fig. 2.22 is also used here for MSE simulation with the NLLS estimator. The variance of the range estimation error is set to be $\sigma^2 = 0.01$, or $\sigma_{i1}^2 = \sigma_{i2}^2 = 0.01$, as shown in Eq. (2.7). The number of iterations in NLLS is set to be $K = 10$.

The simulated MSEs of the proposed scheme for the x- and y-dimensions are shown in Figs. 2.23 and 2.24, respectively, together with the CRLB results. It is observed that the MSE is very close to the CRLB for both the x- and y-dimensions.

Chapter 3: NLOS error mitigation with convex optimization

3.1 Introduction

Localization that uses radio wave as the resources has been widely used in many applications, like the GPS for driving navigation and indoor positioning using Wi-Fi and the UWB signal. Different kinds of localization architecture and techniques have been proposed in literatures that use the radio wave as the resources, including received sgnal strength (RSS) [11], TOA [76], TDOA [77], AOA [60]. Although some of these technologies have been commercialized, many issues remains. One of the key issues is the NLOS problem, which exists in localization systems using radio wave as the resources and severely degrades the localization accuracy.

As shown in Fig. 3.1, the NLOS occurs when the signal in line-of-sight (LOS) direction between the transmitter and receiver is blocked and the signal from the transmitter can only reach the receiver through multipath, like the reflection. Therefore, compared to the LOS scenario, in NLOS scenario, the signal is always delayed in some unknown time, like the Δt in the figure. Since neither the value nor the distribution of the Δt is known, it makes it difficult to estimate it and mitigate its error. This problem has attracted a lot of investigations recently, but majority of these assume that the prior information about the NLOS condition and the error distribution are partially or completely known [48, 97]. In reality, such information is difficult to obtain. Recently, convex optimization techniques have been applied to mitigate the NLOS error without any prior information about NLOS. Interested readers can refer to literature [50–53, 71, 89]. To further mitigate the NLOS error, extra cooperation information between nodes is introduced in [56]. In addition, machine learning is also employed to identify the NLOS links [70].

In this chapter, the newly developed algorithms for the NLOS error mitigation in TOA and TDOA system are introduced. In the first section, the NLOS error

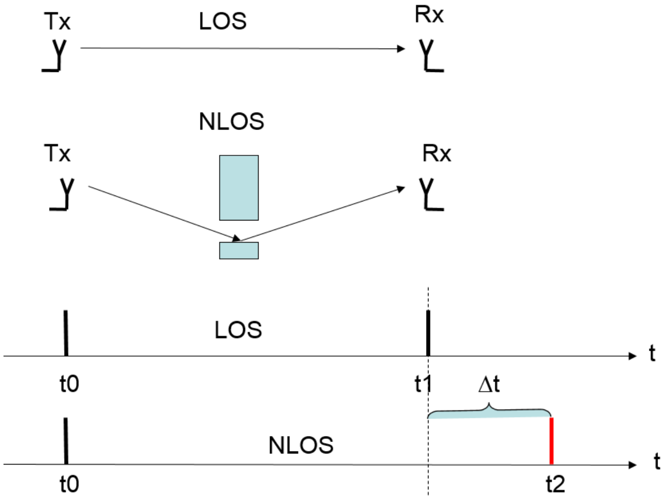


Figure 3.1: The NLOS scenario.

mitigation with convex optimization in TOA system is reviewed and their disadvantages are discussed, like the issue of infeasibility of constrains. To overcome this issue, the soft-minimum constrain is introduced. In the second section, several new schemas to mitigate the NLOS error are discussed. In the third section, the specific difficulties in NLOS error mitigation in TDOA system are discussed. To simplify and effectively reduce the NLOS error mitigation in TDOA system, the TDOA system is first transformed into a TOA system, and then the SDP is applied to mitigate the NLOS error in the transformed system with new constrains. The final section concludes this chapter.

3.2 Revised SDP algorithm for NLOS error mitigation in TOA system

3.2.1 Introduction

Convex optimization has recently been applied for NLOS mitigation in TOA localization without requiring any prior information about NLOS errors [71,75,88–90].

However, convex optimization for NLOS mitigation in TOA systems is often an infeasible problem. To reduce the infeasible problem probability, a main strategy adopted in existing work is to relax (or even eliminate, in some cases) related constraints for the optimization. Such a strategy does not completely resolve the infeasibility issue. Additionally, the positioning accuracy is often reduced because of the relaxed or eliminated constraints applied in the optimization process.

In this section, we develop a soft-minimum method for NLOS mitigation in TOA systems. It has a few useful properties. First, like with existing convex optimization schemes, it does not require any *a priori* information about NLOS links or NLOS error statistics. Second, unlike the convex optimization schemes, it does not have the infeasibility issue. Third, it results in a higher positioning accuracy than with existing convex optimization schemes because of the stricter constraints applied. We will analyze the infeasible problem and develop the proposed soft-minimum method. This method can be applied to any convex optimization algorithms for NLOS error mitigation in TOA systems, but as one example, we will apply it to the SDP algorithm [75, 89, 90] that generally has good performances and is commonly adopted; application of the proposed method to other convex optimization methods such as the second-order cone programming (SOCP) [71,88] follows exactly the same process. Simulation results will be obtained to validate that the proposed method does not have the infeasibility issue and the localization accuracy with it is better than that with existing optimization techniques.

3.2.2 Infeasibility problem in convex optimization

Consider a TOA network with one source node, the target which is a transmitter, and M receivers in an NLOS environment. The range measurement between the source node and receiver i is expressed as

$$r_i = d_i + b_i + n_i, \quad i = 1, 2, 3, \cdots, M \tag{3.1}$$

where d_i is the true distance between the source node and receiver i, n_i is the measurement error, called noise, which has a Gaussian distribution with zero mean and variance σ^2, and b_i represents an NLOS-caused positive bias. Note that r_i can be written as $r_i = ct_i$, where t_i is the signal propagation time from source node to receiver i and c is the speed of light in free space. This relationship allows a practical system to calculate the range measurement by measuring the time spent by a radio frequency signal to travel from the source to the receiver. Also note that if a LOS component exists between the source node and receiver i, then $b_i = 0$. In practical systems, which ones of the M links are NLOS links and the statistics of the corresponding NLOS biases are unknown.

Convex optimization is employed in [71, 75, 88–90] to mitigate the NLOS error in TOA localization systems without requiring prior information about the NLOS links or the NLOS bias statistics. In [71, 88], the second-order cone programming combined with constraint relaxation to mitigate the NLOS error. In [75, 89, 90], an SDP method combined with constraint relaxation is used for NLOS error mitigation.

A common problem of these methods is the infeasibility issue of the convex optimization. A constraint is introduced in these methods:

$$r_i^2 \geq d_i^2 = (\boldsymbol{y}_i - \boldsymbol{x})^T(\boldsymbol{y}_i - \boldsymbol{x}), \tag{3.2}$$

where $\boldsymbol{y}_i$ is the location of the ith receiver and $\boldsymbol{x}$ is the transmitter location being estimated. This constraint indicates that the transmitter should be located within the circle with the ith receiver as the center and r_i as the radius. For NLOS links, it is mostly applicable since the NLOS biases in Eq. (3.12) are positive and typically much larger than the measurement noise n_i [89]. This constraint is also necessary and critical for the convex optimization since it forces the estimated position of the transmitter to be within all circles determined by the anchor receivers and their radii r_i [71, 75, 88–90].

However, the constraint in Eq. (3.2) might not be valid in some LOS conditions, because the LOS links do not have a large NLOS bias and the measurement noise

n_i could be negative. Hence, for LOS links, the position of the transmitter may be located outside of the circle. In such cases, if Eq. (3.2) is included in a convex optimization process as a constraint, the optimization will become an infeasible problem since no such a position can be found to be within all circles formed by the anchor receivers and their radii r_i.

To cope with this situation, the work in [75, 89, 90] tries to relax this constraint by adding some fixed boundaries to reduce the infeasible problem probability. The constraint in Eq. (3.2) is loosened into

$$r_i^2 + u_i \geq d_i^2 = (\boldsymbol{y}_i - \boldsymbol{x})^T(\boldsymbol{y}_i - \boldsymbol{x}). \tag{3.3}$$

This new constraint can be written into an equivalent matrix form via the Schur complement as

$$\begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{y}_i - \boldsymbol{\theta} \\ (\boldsymbol{y}_i - \boldsymbol{\theta})^T & r_i^2 + u_i \end{pmatrix} \succcurlyeq \boldsymbol{0}, \tag{3.4}$$

where $\succcurlyeq$ denotes the matrix inequality, $\succcurlyeq \boldsymbol{0}$ denotes that the matrix on the left-hand side is positive semidefinite and u_i is the fixed boundary [89] expressed as

$$u_i = 4\sqrt{(b_i + d_i)^2\sigma_i^2}. \tag{3.5}$$

A problem of this method is that the infeasibility problem remains, although the probability of encountering an infeasible problem is reduced. Also, the relaxed constraint will lower the estimation accuracy.

A four-step approach is developed in [71, 88] to resolve the infeasibility problem of the convex optimization. In the first step, a strict constraint is used to perform the estimation. In the second step, if the previous estimation is infeasible, the constraint is relaxed and perform the estimation again. In the third step, the constraint is relaxed further if the problem is infeasible in the second step. If the problem is still infeasible, in the fourth step the constraint is eliminated to guarantee that the optimization converges to a solution. Although this method guarantees the convergence of the optimization, it takes several rounds of estimations to obtain a location estimate. Also, if the final step is needed, where the constraint is eliminated to guarantee convergence, the estimation accuracy will be significantly reduced.

3.2.3 Proposed soft-minimum method

The goal of the soft-minimum method being developed in this section is to completely resolve the infeasibility problem of the convex optimization while the constraint in Eq. (3.2) is kept to be optimal.

The basic idea of the soft-minimum method is that, similar to constraint in Eq. (3.3), we also introduce a threshold u_i. The difference is that we set u_i to be variable, rather than a fixed value. Apparently, the larger the variable u_i is, the higher the probability of reaching feasible estimation. For example, as an extreme, if we set the variable $u_i = \infty$, then the optimization will be guaranteed to be a feasible problem. However, if the value of u_i is large, it is virtually equivalent to removing the constraint, which will result in a poor estimation accuracy.

It is thus critical to set an optimal value for u_i in the sense that u_i should be large enough to guarantee the feasibility of the optimization and still be small enough to ensure that the constraint like the one in Eq. (3.3) is as strict as possible. We set the variable u_i as an individual term in the objective function with some proper constant factor μ. This term is written as

$$\mu \sum_{l=1}^{M} u_i^2. \tag{3.6}$$

Take the SDP algorithm in [89] as an example. With some simple revision as discussed above, the SDP optimization problem can be expressed as

$$\operatorname*{argmin}_{\boldsymbol{x},z,h_i,u_i,c_i} \sum\nolimits_{l=1}^{M} w_i(r_i^2 - h_i - c_i)^2 + \sum_{l=1}^{M} \rho c_i^2 + \mu \sum_{l=1}^{M} u_i^2 \tag{3.7}$$

subject to

$$h_i = \begin{pmatrix} \boldsymbol{y}_i \\ -1 \end{pmatrix}^T \begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{\theta} \\ \boldsymbol{\theta} & z \end{pmatrix} \begin{pmatrix} \boldsymbol{y}_i \\ -1 \end{pmatrix}, \tag{3.8a}$$

$$\begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{\theta} \\ \boldsymbol{\theta} & z \end{pmatrix} \succcurlyeq \mathbf{0}, \tag{3.8b}$$

$$\begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{y}_i - \boldsymbol{\theta} \\ (\boldsymbol{y}_i - \boldsymbol{\theta})^T & r_i^2 + u_i \end{pmatrix} \succcurlyeq \mathbf{0}, \tag{3.8c}$$

$$u_i \geq 0, \tag{3.8d}$$

$$c_i \geq 0, \tag{3.8e}$$

where μ is a positive constant whose value need to be determined. Here it is unnecessary to discuss all the details of the SDP algorithm, which can be found in [89]; instead, we focus on the changes made to the algorithm: first, u_i here is a variable, rather than a constant; second, the value of u_i is constrained to be positive as given in Eq. (3.8d), and finally, the individual term $\mu \sum_{l=1}^{M} u_i^2$ is added to the objective function.

With this revision, the constraints in Eqs. (3.8c) and (3.8d) will determine appropriate values for u_i to ensure that the optimization problem is feasible. At the same time, the objective function tries to minimize the whole value for objective function. All the terms in the objective function are positive, the value of the added term $\mu \sum_{l=1}^{M} u_i^2$ will be minimized too. Therefore, the value of u_i is minimized since μ and u_i are all positive.

Another important parameter of this soft-minimum method is the value of the positive constant μ. This factor can be viewed as a weight: a larger μ values means that the added term in Eq. (3.6) will carry more weight in the whole objective function, and therefore u_i can be better optimized. However, if the value for μ is too large, the minimization of the other terms in the objective function will be weakened, and the overall accuracy may be compromised. On the other hand, a very small value for μ may weaken the minimization on the added term and result in a loose constraint in Eq. (3.8c). Therefore, a proper value for μ should be determined first and typically this searching process can be done via simulation.

In this SDP algorithm, the objective function happens to be a minimization problem. If it is a maximization problem, we can simply add the term $-\mu \sum_{l=1}^{M} u_i^2$ in the objective function. In the discussion above, although the SDP algorithm is used as an example, the process is exactly the same if the proposed method is applied in other convex optimization schemes such as the SOCP [71, 88].

3.2.4 Simulation and results

3.2.4.1 Simulation setup

In this section, the performance of the proposed soft-minimum method applied in the SDP algorithm is evaluated via simulation and is compared with that of the original SDP algorithm. In the simulation, the u_i is set in two different cases: a) u_i for each receiver, or each link, is different, and b) $u_i = u$, where u is a constant, for all receivers. The estimators considered are listed in Table 3.5.

Estimator	Description
SDP	The original SDP algorithm for NLOS error mitigation in TOA localization system described in [89].
P_1	SDP algorithm with the proposed soft-minimum method with different u_i values for each link.
P_2	SDP algorithm with the proposed soft-minimum method with a fixed value of u_i for all links, that is, $u_i = u$.
ML-LOS	The ML estimator assuming LOS links only as described in [89].

Table 3.1: Estimators considered.

In order to obtain the best performance of the proposed method, a proper value for μ should be determined. To this end, several different value for μ within the range from 10 to 1000 are simulated. Also, the proposed method is simulated under different noise conditions, namely, the standard deviation (std) of the measurement noise n_i is set as 0.1, 0.2, 0.3, 0.4 and 0.5 meters, respectively.

The configuration of the simulation is shown in Fig. 3.19, where eight anchors and one source node is considered. In the simulation, 49 locations of the source node are considered and they are uniformly distributed in a $3m \times 3m$ space. The NLOS bias is generated according to a uniform distribution between value of $[0m, 2m]$. Note that the bias information is unknown to all estimators in the simulation. The weighting elements, w_i are all set to 1; that is, all links are weighted equally. The penalization factor ρ in Eq. (3.7) is set to $\rho = 0.1$ as in [89].

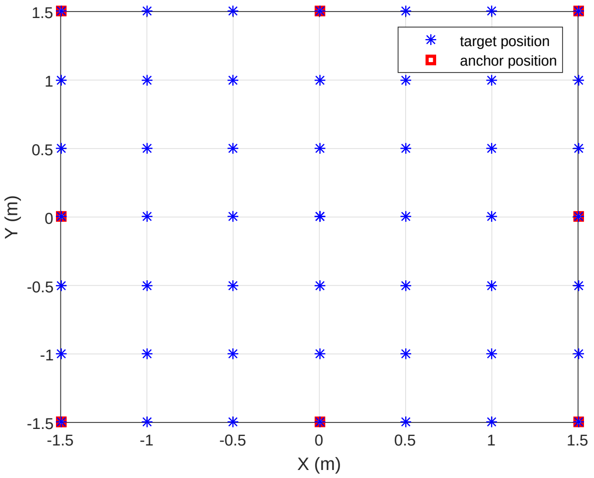


Figure 3.2: Simulation configuration.

For the original SDP algorithm, Eq. (3.5) is applied. But for the proposed method, u_i is a variable. The maximal likelihood (ML) estimator assuming only LOS links (ML-LOS), which gives the best performance potentially achievable, is used as the benchmark estimator. Both of the proposed scheme and original SDP estimators are solved with SDPT3 in Matlab CVX toolbox.

3.2.5 Simulation results

The average feasible optimization percentages of the SDP algorithm and the proposed method are listed in Table (3.6) under different noise conditions. Note that the percentage of feasibility is calculated as the number of feasible cases over all simulated cases. It is observed from the simulation results that for all noise conditions, optimization with the proposed methods, P_1 and P_2, is feasible. For the SDP algorithm, about 4% $\sim$ 5% of cases are infeasible; that is, the minimization does not converge when the boundary condition expressed in Eq. (3.5) is applied.

Noise std (m)	0.1	0.2	0.3	0.4	0.5
SDP feasibility (%)	96.1	95.5	95.5	95.4	95.4
P_1 feasibility (%)	100	100	100	100	100
P_2 feasibility (%)	100	100	100	100	100

Table 3.2: Average feasible optimization percentages of SDP and the proposed methods under different noise conditions.

The positioning mean-squared error (MSE) with the SDP and the proposed methods under different noise conditions are shown in the Fig. 3.3. The first observation we can make from this figure is that the performances of both cases of the proposed method, P_1 and P_2 as described in Table 3.5, are better than that of the original SDP estimator, because the proposed method applies a stricter constraint u_i expressed in Eq. (3.8d). This result demonstrates that the proposed method is able to automatically determine a smaller u_i than the fixed u_i in the original SDP estimator to form a stricter constraint. The second observation we can make from the results in this figure is that, for both cases of the proposed method, P_1 and P_2, when noise is low, cases with a lower value of μ tend to outperform cases with a larger value of μ. For example, consider the case when std $= 0.1m$: the estimator with $\mu = 10$ outperforms the estimator with $\mu = 1000$ for both P_1 and P_2. The situation is reversed when noise error is high; for example, when std $= 0.5m$, the estimator with $\mu = 1000$ outperforms the estimator with

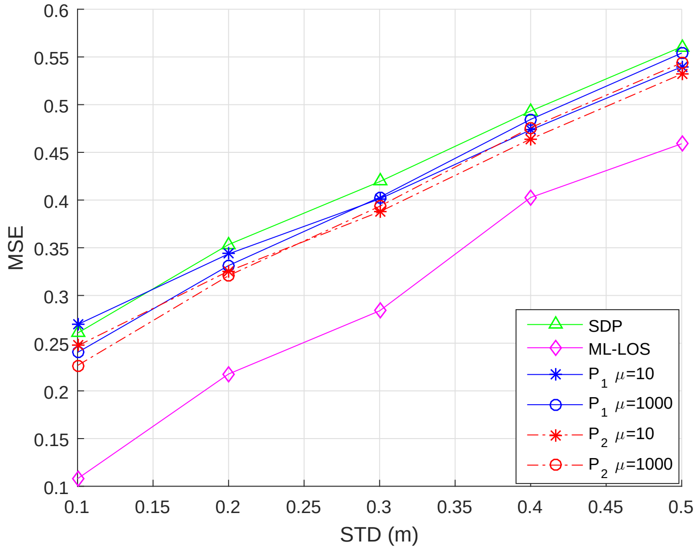


Figure 3.3: Simulated MSE results of SDP with the proposed soft-minimum method and the original SDP estimator (for clarity, only cases with $\mu = 10$ and $\mu = 1000$ are shown).

$\mu = 10$. Therefore, to achieve a better performance, in high-noise situations, a relative smaller μ should be used and vice versa. The third observation we can make from these results is that the performance of P_2 is better than that of P_1, although both P_1 and P_2 outperform than the original SDP estimator.

In summary, the proposed methods are able to completely remove the infeasibility problem and also improve the localization accuracy.

3.3 NLOS error mitigation with a bias iterative deduction scheme in TOA based localization

3.3.1 Introduction

In last section, the revised SDP algorithm was proposed to mitigate the NLOS error in the TOA system. Without additional information, it's difficult to further mitigate the error, based on the algorithm itself. Since additional information or prior information for NLOS, like NLOS links and NLOS error statistics, are in general unavailable, it's difficult to further mitigate the error.

In this and the next sections, two schemes are proposed to try to discover some 'additional' information based on estimation. Based on the estimation, we try to predict which links are in NLOS and how much bias the NLOS contributes to the NLOS link. Then this predicted information is used to further mitigate the NLOS error.

3.3.2 Motivation

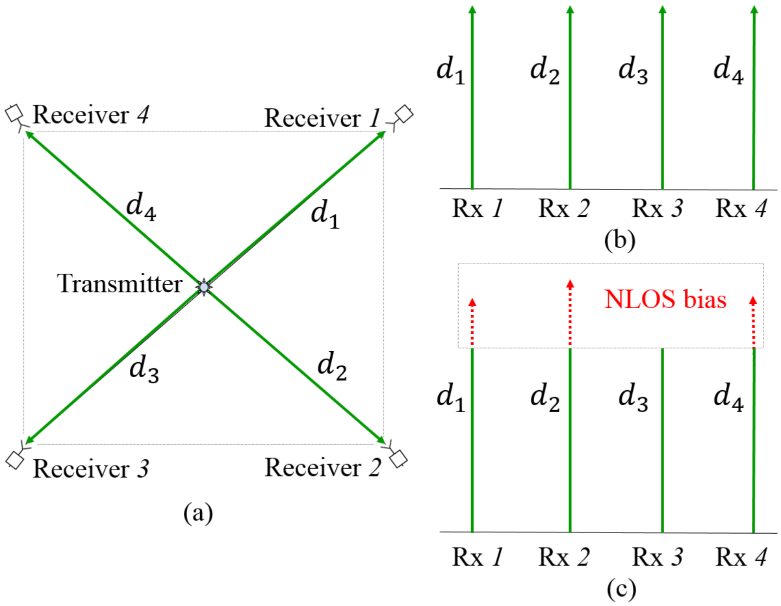


Figure 3.4: LOS/NLOS condition in TOA model: (a) TOA model; (b) Observations in LOS condition; (c) Observations in NLOS condition.

The TOA model with one source node and M receivers in the NLOS environ-

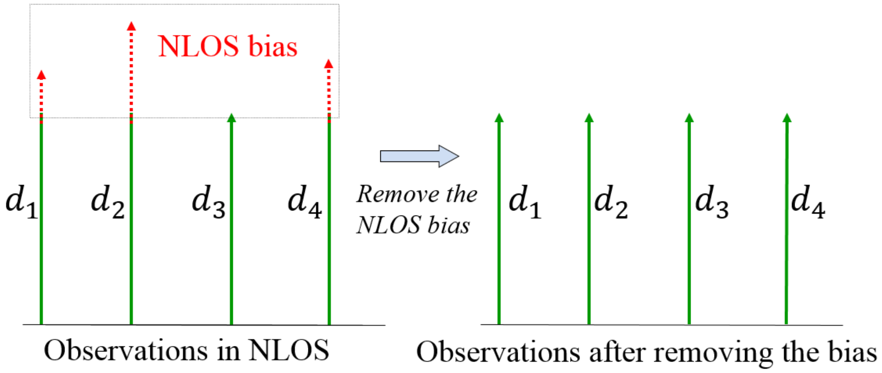


Figure 3.5: Proposed scheme with bias deduction.

ment can be shown as

$$r_i = d_i + b_i + n_i, i = 1, 2, 3, ..., M, \tag{3.9}$$

where $r_i = ct_i$ is the measured distance between the source node and the i^{th} receiver, t_i is the signal propagation time from the transmitter to the i^{th} receiver, c is the speed of light in free space, and d_i is the true distance between them. The b_i represent the positive bias caused by the NLOS effect, while n_i is the noise, which is usually modeled as a Gaussian distribution with a zero mean and a variance σ^2.

The NLOS occurs when the LOS direction between the source node and the anchor node is blocked. In such conditions, the signal will be only able to arrive at the receiver through several multipaths. As the traveling distance from the multipath is always longer than that from the LOS route, the time spent in the multipath route should be more than that in the LOS route. Therefore, the NLOS condition always introduces a positive bias into the TOA model. As shown in Fig. 3.4, a simple TOA-based localization example with four anchors (Rx1$\sim$Rx4) on each corner of a square and a transmitter in the center, the four observed distances ($d_1 \sim d_4$) should be identical in LOS condition if noise is not considered. But in the NLOS condition, the observed distances are lengthened as the additional NLOS bias is added onto the original value, as shown in Fig. 3.4.b and 3.4.c.

Since the NLOS bias is always positive, the proposed idea tries to remove the bias from the observations in NLOS links, as shown in figure 2. After removing the bias, the revised observations without bias can be used to do the location estimation. So, the key problem for this scheme is how to remove the NLOS bias without any *a priori* information about NLOS, or specifically: 1) How to identify which links are in NLOS condition, and 2) How to obtain the bias values for the identified NLOS link. For problem 1), in previous literature, several proposed methods are used to identify the NLOS links with some prior information [80, 97]. But they are not applicable here, as the prior information about NLOS is assumed to be unknown. For problem 2), if the bias has already been exactly known, we should be able to estimate the position precisely and don't need extra mitigation operations. Therefore, based on previous techniques, it's difficult to identify all NLOS links and know the value of bias for each NLOS link.

Without any prior information, we try to identify the NLOS link and estimate the bias needing to be removed from an observation based on the estimation. Since the estimation itself is not accurate, any decision based on it is not accurate, either. To increase the correctness of the judgments, instead of trying to identify all NLOS links, we simplify the problem to locate the link with the biggest estimated bias and try to remove the bias iteratively.

3.3.3 Proposed scheme

In proposed scheme, several measures are considered and the general flow of the procedure is shown in Fig. 3.12. The ***first*** measure of this scheme is the estimator used to estimate the bias. Apparently, the better the accuracy of the estimator is, the higher the correctness of judgments will be in this procedure, because all judgments are based on the results from this estimation. The ***second*** measure is to trust the estimated bias partially (e.g., with 30% of trust). Since the estimator is not 100% accurate, the bias estimated from it is not accurate either. However, depending on the estimator used, we can trust the estimated bias to a certain degree. For example, we just remove $0.3 \times b$ from the observation, where b is the estimated bias with selected estimator. The ***third*** measure is to do the reduction iteratively until some threshold is met. Because each time we just remove part of the estimated bias, we cannot finish this procedure in single step. Instead, we need to repeat the deduction and stop it until some threshold is met. The ***fourth*** measure is to consider how many NLOS links to be processed in each iteration.

Among the above measures, there are some trade-offs between them. For example, for the second measure, if we set the credibility to a very low level, e.g., 1%, the correctness of the bias deduction will be higher than that when it's set to, 90%, for example. But it will greatly increase the repetition times in the third measure and therefore increase the total computation time, and vice versa. The situation is also true for the third and fourth measures.

So, based on the above analysis and assumptions, the general bias deduction scheme can be shown in algorithm (3), where $\boldsymbol{r}$ and $\boldsymbol{b}$ are vectors with $\boldsymbol{r} = (r_1, r_2, ..., r_M)$, $\boldsymbol{b} = (b_1, b_2, ..., b_M)$, $b_{\max}$ is the maximal bias in $\boldsymbol{b}$, ϵ is a threshold used to judge whether a link is in NLOS or not and β is a factor with $\beta \in [0, 1]$.

To improve the correctness of the judgments, we can revise only one observation with the biggest estimated bias in each iteration, instead of modifying all of them, or

$$r_{b_{\max}} := r_{b_{\max}} - \beta b_{\max}, \tag{3.10}$$

where $b_{\max} = max(\boldsymbol{b})$ and $r_{b_{\max}}$ represents the related observation with the biggest estimated bias. And the parameter β can be set to different values according to the value $b_{\max}$. For example, we can set $\beta = 0.6$ when the estimated $b_{\max} > 1$ meter

Algorithm 1 : General bias deduction scheme

```
Input: $\boldsymbol{r} = (r_1, r_2, ..., r_M)$,
Repeat until $(b_{\max} < \epsilon)$
    Estimate $\boldsymbol{b}$ with an estimator,
    Revise $\boldsymbol{r}$ as: $\boldsymbol{r} := \boldsymbol{r} - \beta\boldsymbol{b}$,
Estimate with the final revised observation $\boldsymbol{r}$.
```

and $\beta = 0.2$, when $b_{\max} < 0.2$ meter, based on the fact that the larger estimated bias has a higher possibility to be a NLOS link. To prevent some special diverging case where the estimated position may go to infinity, an additional constraint is set as $b_{\max} < b'_{\max}$, where $b'_{\max}$ is the $b_{\max}$ in the previous iteration. The revised bias deduction scheme is shown as

Algorithm 2 : Revised bias deduction scheme

```
Input: $\boldsymbol{r} = (r_1, r_2, ..., r_M)$,
Repeat until ($b_{\max} < \epsilon$  &&  $b_{\max} < b'_{\max}$)
    Estimate $\boldsymbol{b}$ with an estimator,
    $b'_{\max} = b_{\max}$,
    $\beta = f(b_{\max})$,
    Revise $r_{b_{\max}}$ as: $r_{b_{\max}} := r_{b_{\max}} - \beta b_{\max}$,
Estimate with the final revised observation $\boldsymbol{r}$.
```

In this algorithm, $f(\cdot)$ is a mapping function that is set as an error function (ERF)

$$\beta = \operatorname{erf}\left(\frac{b_{\max}}{\gamma}\right), \tag{3.11}$$

where γ is a parameter used to adjust the inclination of the function. In this simulation, γ is set to be 1. As $b_{\max}$ is always larger than 0, β in Eq. (3.11) will take a value between zero and one.

3.3.4 Simulation and results

3.3.4.1 Simulation setup

In this section, the performance of the proposed scheme is evaluated by applying to the state-of-art NLOS mitigation estimator in TOA localization.

The configuration of the simulation is shown in Fig. 3.6, where eight anchors and one source node are considered. In the simulation, 36 locations of the source

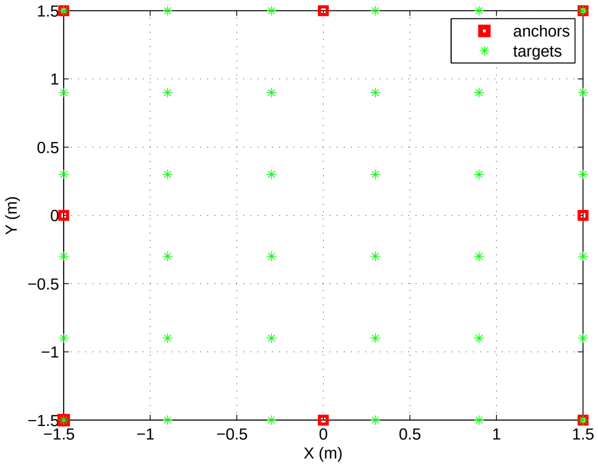


Figure 3.6: Simulation configuration.

node are considered and they are uniformly distributed in the $3m \times 3m$ space. The NLOS bias is generated according to a uniform distribution between a value of $[0m, 2m]$. Note that the bias information is unknown to all estimators in the simulation.

In the simulation, the state-of-the-art SDP estimator in [89] is chosen to be the estimation algorithm in the scheme. For the SDP estimator, we set the weighting elements to be $w_i = 1$, as we assume that the measured distance is only disturbed by the Gaussian noise and the NLOS effect. And the noise n_i is Gaussian distributed with zero mean and $\sigma_2 = 0.01$. The penalization factor is set to be $\delta = 0.1$, and μ_i is set as $\mu_i = 3\sigma$, according to the literature.

In the simulation, the scheme in algorithm 2 is used. The number of anchors is $M = 8$ and the threshold ϵ is set as $\epsilon = 0.1$.

3.3.4.2 Simulation results

To better understand the performance of the proposed scheme, the results for the first five iterations (labeled as P1 to P5) and the final iteration (labeled as P-final) are shown. Also, it would be informative to compare them with the benchmark estimator: the Maximal Likelihood (ML) estimator using only LOS links (ML-LOS), which stands for the potential best performance and is only used for comparison. In addition, two other estimators are considered: the SDP-ROBUST estimator [71] and the ML with all links (ML-ALL), as shown in Tab. (3.5).

The simulation results are shown in Figs. 3.15 – 3.17, which use the cumulative distribution function (CDF) of the location errors for the considered estimators. Fig. 3.15 shows the results in light NLOS condition when there are 1 or 2 NLOS links among eight total links. In this scenario, from the second iteration $p2$, the proposed scheme starts to outperform the SDP estimator. But the performance after the third iteration almost stays the same. This is because in light NLOS condition, it is enough to mitigate the bias in only one or two links in the first few iterations.

In Fig. 3.16, when the number of NLOS links is 3, 4, or 5, the proposed scheme starts to outperform the SDP estimator from the third iteration $p3$, and each iteration provides some improvements. In light NLOS conditions, therefore, more iterations are required to mitigate the bias when more links are in the NLOS condition.

In severe NLOS condition, when there are 6, 7, or 8 NLOS links, the results are shown in Fig. 3.17, where the ML-LOS is not applicable, as less than three LOS links are available to do a 2-dimensional estimation. In this case, the proposed method can't improve the performance over the SDP estimator. However, the SDP-ROBUST estimator starts to provide reasonable performance. The ML-ALL still has the worst performance.

In this simulation, the average number of iterations for all conditions is around 12 for the proposed scheme.

In general, when applied to the SDP estimator, the proposed scheme further improves the performance in light and median NLOS conditions over that of the SDP estimator.

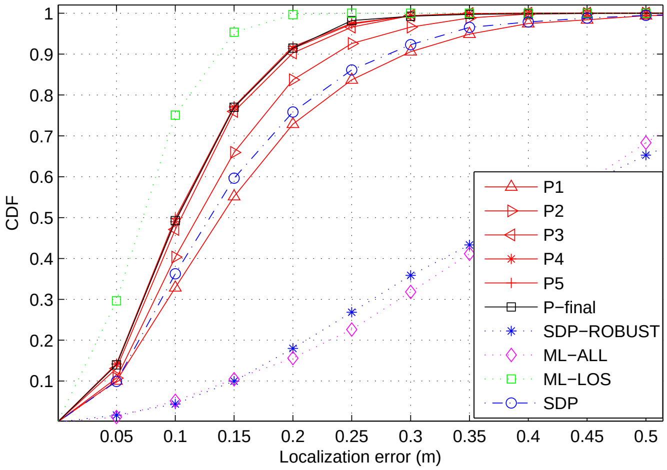


Figure 3.7: Estimation error CDF with light NLOS conditions (NLOS=1 or 2).

3.4 NLOS link identification and discard scheme in TOA based localization

In this subsection, another scheme is introduced to further mitigate the NLOS error.

Estimator	Description
Px	The proposed method with x iterations
P-final	Proposed method in the final iteration
SDP-ROBUST	The robust estimator in [71]
SDP	The SDP estimator in [89]
ML-LOS	The ML estimator with only LOS links in [89]
ML-ALL	The ML estimator with all links in [89]

Table 3.3: The considered estimators.

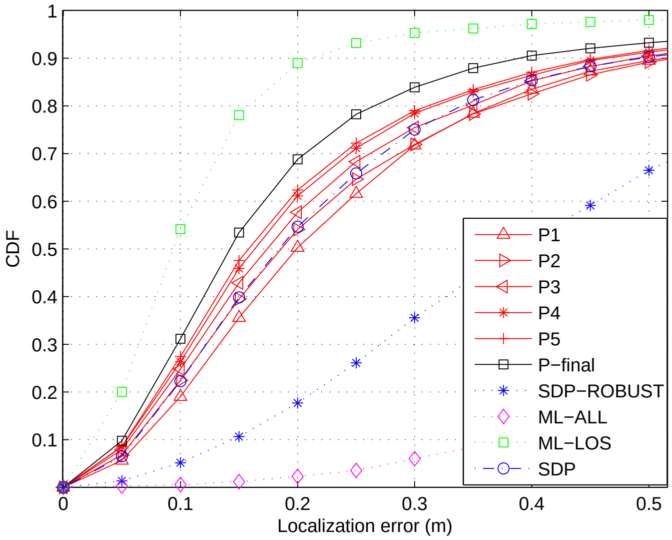


Figure 3.8: Estimation error CDF with median NLOS conditions (NLOS=3, 4 or 5).

3.4.1 Motivation

In the NLOS condition, the unknown bias b_i, as shown in Eq. (3.9), introduces large errors into the localization calculation. As shown in Fig. 3.10, a simple TOA-based localization with five anchors ($Rx1 \sim Rx5$) on the corner of a pentagon and a transmitter in somewhere inside the boundary of the five receivers. The observations in LOS condition can be seen in Fig. 3.10(b), shown as ($d_1 \sim d_5$). But in the NLOS condition, e.g., the fourth and fifth links are blocked by some obstacles and the observed distances are lengthened, since an additional NLOS bias is added, as shown in Fig. 3.10(c). Since the NLOS links introduce large errors into the estimation, the proposed method tries to discard the NLOS links and uses only the LOS links to do the estimation, as shown in Fig. 3.11; the fourth and fifth NLOS links are discarded and only the first three LOS links are used.

So the key problem here is how to identify the NLOS links. In existing literature, prior information about NLOS is required to identify the NLOS links [63,78].

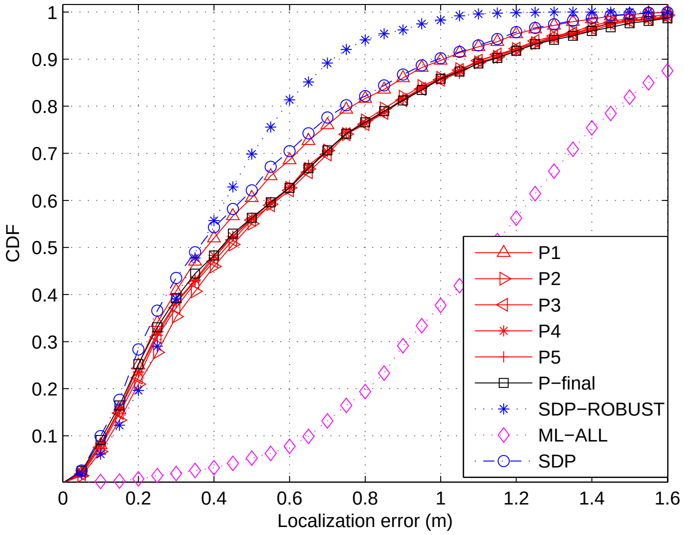
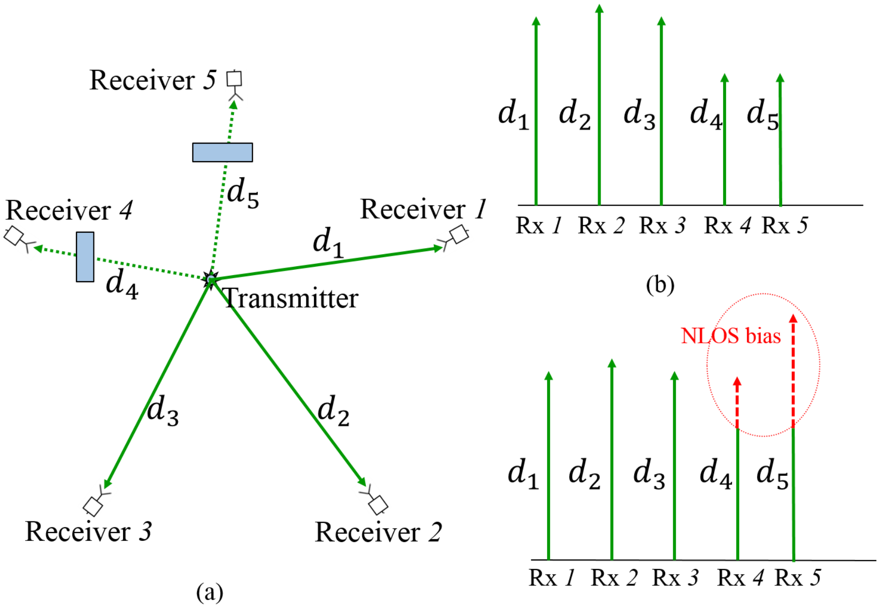


Figure 3.9: Estimation error CDF with severe NLOS conditions (NLOS=6, 7 or 8).

Other methods, like machine learning [70], are also introduced to identify the NLOS links. Here, we introduce a simple but effective scheme to identify and discard the NLOS links by estimating the bias of each link with a location estimator. Since the estimator is not 100% accurate, the identification of NLOS based on the estimated bias is not 100% accurate, either. Although it's difficult to correctly identify all NLOS links, it's still useful if we can eliminate some of the severe NLOS links.

3.4.2 Proposed method

Like the analysis in the last section, the focus of this scheme is to improve the correctness of the NLOS identification and effectively discard the severe NLOS links. In this section, a simple and effective scheme with several considerations is proposed to identify and discard the NLOS links as shown in Fig. 3.12. The ***first*** measure is the localization estimator used to estimate the bias. Apparently, the better the accuracy of the estimator, the higher the correctness of judgments.



(a) (b) (c)

Figure 3.10: System model.

The ***second*** measure is to trust the estimated bias partially (e.g., only the link with the biggest estimated bias is considered and judged whether it's a NLOS link or not). The ***third*** measure is to introduce iterative processing, in which only one link is discarded in one iteration. The iteration will stop when the maximum estimated bias is smaller than a threshold or with the minimal expected number of links is reached.

As shown in Fig. 3.12, the position is estimated by some estimator with a set of observation $\boldsymbol{r}$ in the NLOS scenario. Then the biases $\boldsymbol{b}$ is calculated from the estimated position. If the maximal estimated bias is larger than a threshold ξ and the number of links is larger than the minimal number of links N, the link with the biggest bias will be considered as a severe NLOS link and then be discarded. The iteration will stop when one of above conditions fails and the final position is obtained from the last estimation. In this procedure, several factors will affect the final accuracy: the estimator used, the threshold ξ and the minimal number of

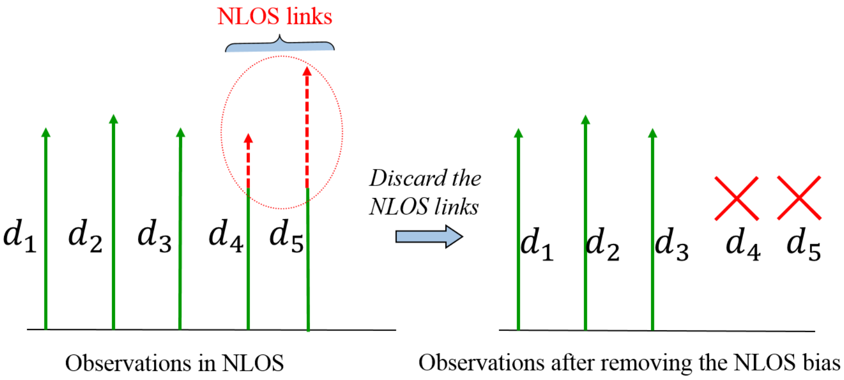


Figure 3.11: Proposed method with NLOS links discarded.

links N. The code form of the procedure can be described as Algorithm 3. In this

Algorithm 3 : General NLOS link discard scheme

```
Input: $\boldsymbol{r} = (r_1, r_2, ..., r_M)$,
Estimate $\boldsymbol{b}$ with an estimator,
Repeat when $(b_{\max} > \xi \quad \&\& \quad n > N)$
    Discard the link $r_{b_{\max}}$
    Estimate $\boldsymbol{b}'$ with left links $\boldsymbol{r}'$,
Estimate with the final left observation $\boldsymbol{r}'$.
```

algorithm, the initial $\boldsymbol{r}$ and $\boldsymbol{b}$ are vectors $\boldsymbol{r} = (r_1, r_2, ..., r_M)$ and $\boldsymbol{b} = (b_1, b_2, ..., b_M)$; $b_{\max}$ is the maximal bias, ξ is a threshold used to judge whether a link is NLOS or not, N is the minimal number of links expected to keep, and $r_{b_{\max}}$ is the link with the maximal bias.

Similar to the simulation results shown in [9], the bias iterative deduction scheme provides better NLOS error mitigation performance in light NLOS conditions than that in median and severe NLOS conditions. It's possible to transform the median NLOS conditions into light NLOS conditions by discarding some of the NLOS links using the proposed scheme. Based on this idea, we propose a method that combines the proposed scheme with the one in [9]. The combined scheme is shown as Algorithm 4. As [9] indicated, β will take the value between zero and

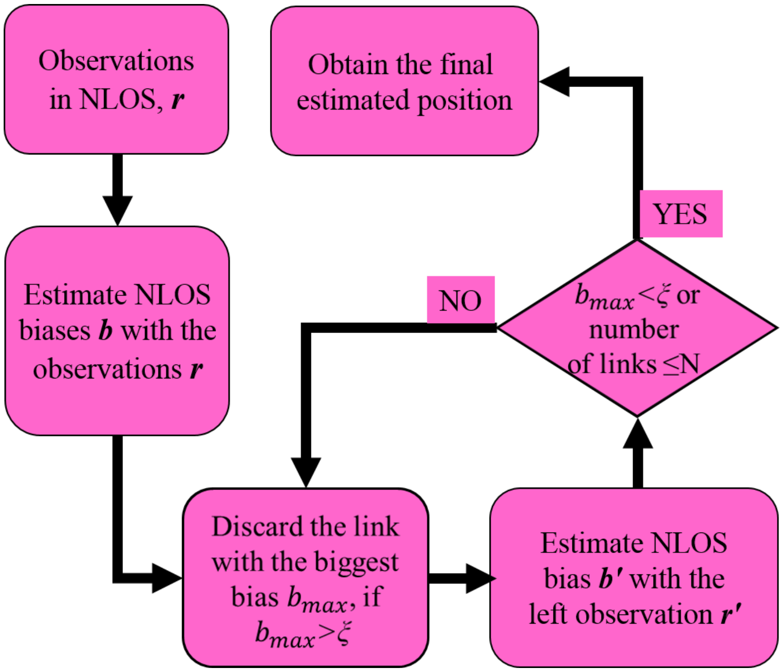


Figure 3.12: Procedure for the proposed method. $\boldsymbol{r}$ and $\boldsymbol{b}$ are observation and related bias vectors, respectively. $b_{\max}$ is the maximal bias, ξ is a threshold, and N is the minimal number of links expected to keep.

one, $b'_{\max}$ is the $b_{\max}$ in previous iterations, and $f(\cdot)$ is a mapping function that is set as an error function (ERF).

3.4.3 Numerical results

3.4.3.1 Correctness of discarded links to be NLOS links

It would be informative to calculate the correctness of the discarded links to be NLOS links. For this, consider the specific setting in Fig. 3.13, where eight anchors and 36 uniformly distributed source locations are considered. The NLOS bias b_i in Eq. (3.9) is generated according to a uniform distribution between the values of $[0m, 2m]$; the variance of the noise is $\sigma^2 = 0.01$.

In the simulation, the state-of-art SDP NLOS error mitigation algorithm for

Algorithm 4 : Combined scheme

```
Input: $\boldsymbol{r} = (r_1, r_2, ..., r_M)$,
Estimate $\boldsymbol{b}$ with an estimator,
Repeat when $(b_{\max} > \xi \quad \&\& \quad n > N)$
    Discard the link $r_{b_{\max}}$
    Estimate $\boldsymbol{b}'$ with with left links $\boldsymbol{r}'$,
Estimate with the final left observation $\boldsymbol{r}'$.
Input: $\boldsymbol{r}'$,
Repeat until $(b_{\max} < \epsilon \quad \&\& \quad b_{\max} < b'_{\max})$
    Estimate $\boldsymbol{b}$ with an estimator,
    $b'_{\max} = b_{\max}$,
    $\beta = f(b_{\max})$,
    Revise $r_{b_{\max}}$ as: $r_{b_{\max}} := r_{b_{\max}} - \beta b_{\max}$,
Estimate with the final revised observation $\boldsymbol{r}'$.
```

TOA localization in [89] is used as the estimator. Since we assume that the measured distance is only disturbed by the Gaussian noise and the NLOS effect, the weighting elements for SDP is set as $w_i = 1$. The penalization factor is set as $\delta = 0.1$, according to the literature, and the compensation parameter is set to $\mu_i = 3\sigma$.

In this calculation, when one link is discarded, we then check whether or not this link is in NLOS. Since the distance, r_i, in Eq. (3.9) is disturbed both by Gaussian noise and NLOS effects, it's reasonable to judge it to be a NLOS link as long as the final error of r_i is larger than some threshold. For example, when a link with $b_i + n_i > t$, where t is some threshold and $t > 0$, it will be considered a NLOS link. Here, three different thresholds are considered: $t = 2\sigma$, 3σ, and 4σ or $t = 0.2m$, $0.3m$, and $0.4m$.

Fig. 3.14 describes the correctness for the nth discarded links, where at most five links are allowed to be discarded. As expected, the situation when $t = 0.2m$ has the highest correctness, which are all above 95%, while the situation when $t = 0.4m$ has the lowest correctness, with all above 93.3%. Also, the first discard has the highest correctness – all of them are above 99.5%. The correctness drops with the increase of n and the number of discarded links.

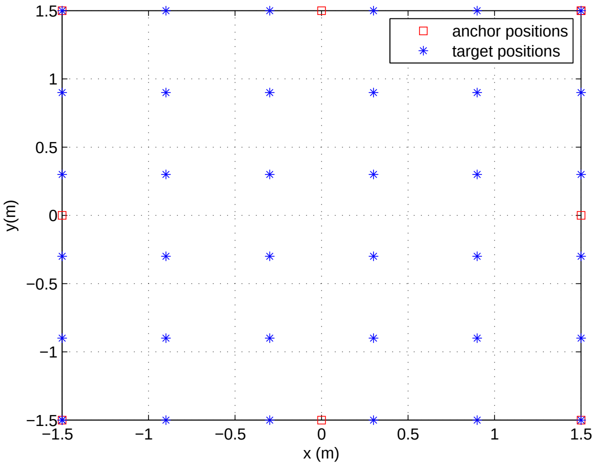


Figure 3.13: Simulation configuration.

3.4.3.2 Performance of the proposed scheme in NLOS mitigation

In the simulation, the performance of the proposed scheme is compared with that of the SDP. In this simulation, the setting in Fig. 3.13 is used. To better understand the performance of the proposed scheme, N in Algorithm 1 is set as $8 \sim 4$, or the maximal number of links can be discarded is $1 \sim 4$, respectively, and $\xi = 0.4m$.

The results are shown in Fig. 3.15 to Fig. 3.17, where the cumulative distribution function (CDF) of the location error is used for the considered estimators, as shown in Table 1. The NLOS scenarios are divided into light, median, and severe NLOS conditions with (1, 2), (3, 4, 5), and (6, 7, 8) NLOS links out of the total eight links, respectively.

Fig. 3.15 shows the results in the light NLOS condition, when the number of NLOS links is 1 or 2. As expected, the ML estimator using only LOS links (ML-LOS) has the best performance. However, it is only used as a benchmark for comparison, as we assume the estimators don't know which links are in the LOS condition. The curves labeled as $P_4 \sim P_8$ represent the condition with $N = 4 \sim 8$, respectively, or at most $4 \sim 0$ links can be discarded. In this condition, P_7 and P_6 greatly improve the performance, while P_4 and P_5 keep the same performance as

P_6. These results demonstrate that P_7 and P_6 can correctly discard the link with large bias to improve the performance compared to the SDP estimator.

In Fig. 3.16, the median NLOS condition, when NLOS=3, 4, or 5 of the total eight links, is considered. From the results, starting from P_6, the performance of the proposed scheme starts to outperform that of the SDP estimator, and P_4 has the highest performance. Fig. 3.17 shows the performance in the severe NLOS condition, where less than three LOS links exist and, therefore, ML-LOS doesn't apply. In such a condition, when the majority of links are in the NLOS condition, discarding the NLOS links doesn't help to mitigate the NLOS error.

Compared with the SDP estimator, the proposed scheme can effectively improve the performance in light to median NLOS conditions.

3.4.4 Performance of the combined scheme

In this simulation, the setting in Fig. 3.13 is used again. To better understand the performance of the combining scheme, we apply the NLOS bias deduction scheme to all NLOS conditions with the setting that N in Algorithm 1 is set from 8 to 5, respectively. Both schemes use the SDP estimator as the estimation algorithm.

The results are shown from Fig. 3.15 to Fig. 3.17, where C_x stands for the combined scheme which should keep at least x links, or $N = x$. To compare the performance of the proposed scheme with the bias deduction scheme, we can refer to P_4, which stands for the best performance of the proposed scheme and C_8, which stands for the performance of the NLOS bias deduction scheme in [9]. From the results, the two have the similar performance, but the average running time for C_8 is about 12 iterations of SDP estimation while P_4 needs about three iterations, which is only 25% of the iterations required by C_8.

To see the performance of the combined scheme in Fig. 3.15 to 3.17, we can compare the performance of P_x and that of C_x, with at most x links discarded. As expected, in the median NLOS condition, the combined scheme can further improve the performance over both of proposed scheme and the bias deduction scheme in [9].

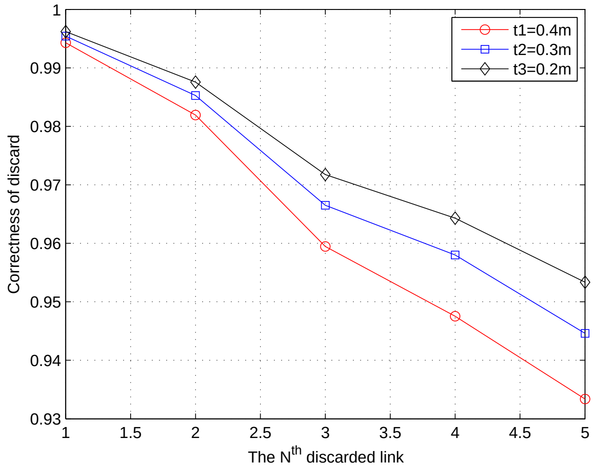


Figure 3.14: Correctness of discard for the nth discarded links in the whole simulation, where 1=100%.

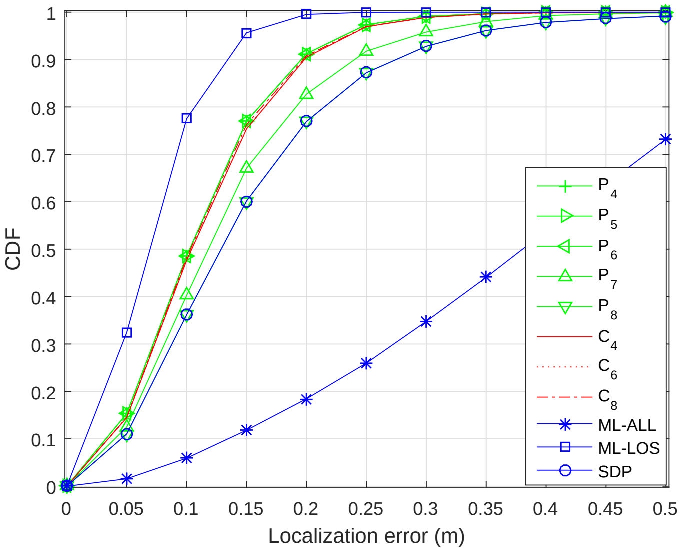


Figure 3.15: Estimation error CDF with light NLOS conditions (NLOS=1 and 2).

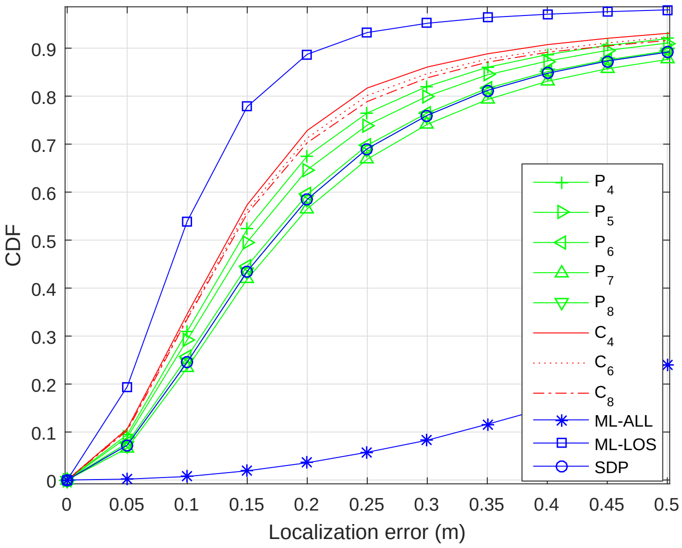


Figure 3.16: Estimation error CDF with median NLOS conditions (NLOS=3, 4 and 5).

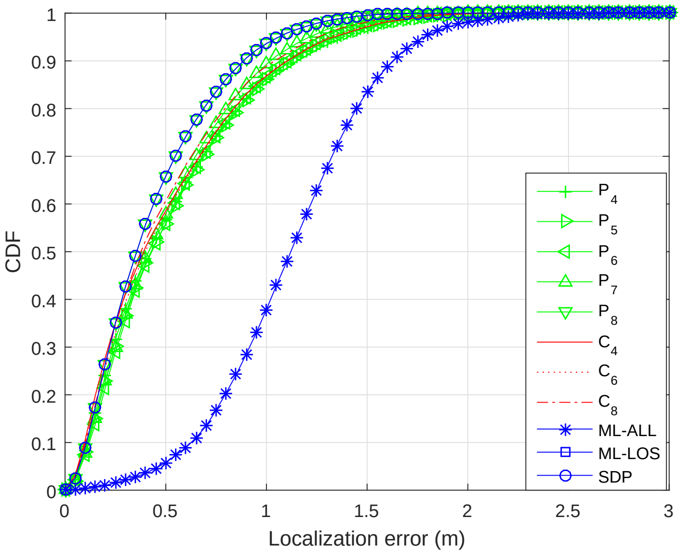


Figure 3.17: Estimation error CDF with severe NLOS conditions (NLOS=6, 7 and 8).

Estimator	Description
P_x	The proposed scheme with at least x links should be kept or $N = x$
C_x	The combined scheme with at least x links should be kept or $N = x$. [9]
SDP	The SDP estimator in [89]
ML-LOS	The ML estimator with only LOS links in [89]
ML-ALL	The ML estimator with all links in [89]

Table 3.4: The considered estimators.

3.4.5 Conclusion

In this section, a NLOS discard scheme to mitigate the NLOS error in TOA localization is introduced. The basic principle of the proposed scheme and its procedure are presented in detail. To evaluate the performance of the proposed scheme, the correctness of the discarded links to be NLOS links is calculated, which shows that the probability the discarded links are NLOS links is very high. From the simulated CDF results, it's found that the proposed scheme provides better performance than the state-of-art SDP algorithm in light to median NLOS conditions. Although the proposed scheme presents similar CDF performance to that of the bias iterative deduction scheme, it uses only 25% of the iterations compared to the bias iterative deduction scheme. In addition, the simulated CDF results also show that the combined scheme can further improve the performance in median NLOS conditions.

3.5 NLOS error mitigation in the TDOA system

3.5.1 Introduction

NLOS error mitigation and NLOS link identification for TOA localization have been investigated extensively, assuming that *a priori* information of NLOS links and/or NLOS error statistics is available [78–82], or other forms of system/channel resources are available [83–87]. Convex optimization has recently been applied for NLOS error mitigation in TOA systems and achieves a good performance without requiring such information [88–90].

Convex optimization for NLOS error mitigation in TDOA systems is also extensively investigated [91–96]. Issues remain unsolved of this technique for TDOA systems include the convex hull problem, proper reference-anchor selection, and difficulties dealing with a wider range of NLOS-caused ranging errors. In TOA systems, the NLOS error is always positive, and is typically larger than the measurement noise. Hence, restricting the target to be in a circle formed by the anchor and the target's observation will improve the localization accuracy due to the stricter constraint. In TDOA localization, however, the NLOS-caused ranging errors could be positive or negative, making the problem complex. Also, convex optimization for TDOA systems might encounter the convex hull problem; that is, the estimated positions always lie in the convex hull formed by the sensor nodes [94]. In addition, since a reference must be selected first to obtain the observations, it is possible that an NLOS link might be chosen as the reference, which will further degrade the localization performance. To avoid choosing an NLOS link as the reference, different methods have been developed: in [92] a method that tries each anchor as the reference to estimate the location and uses some linear combination of the estimated locations as the final results is developed; in [94] an LOS link is identified first, which is then used as the reference for position estimation.

These methods result in very complex estimators, and still do not effectively solve all the problems. Also, methods developed for NLOS mitigation in TOA localization cannot be applied directly to TDOA systems due to their architectural differences and the issues discussed above [91].

In this section we propose a new technique that transforms a TDOA architecture into a TOA architecture for NLOS mitigation in TDOA systems, and develop a semidefinite programming method for this technique. The proposed technique has has several advantages over existing convex optimization schemes. First, it effectively resolves the aforementioned issues that existing schemes are facing for NLOS

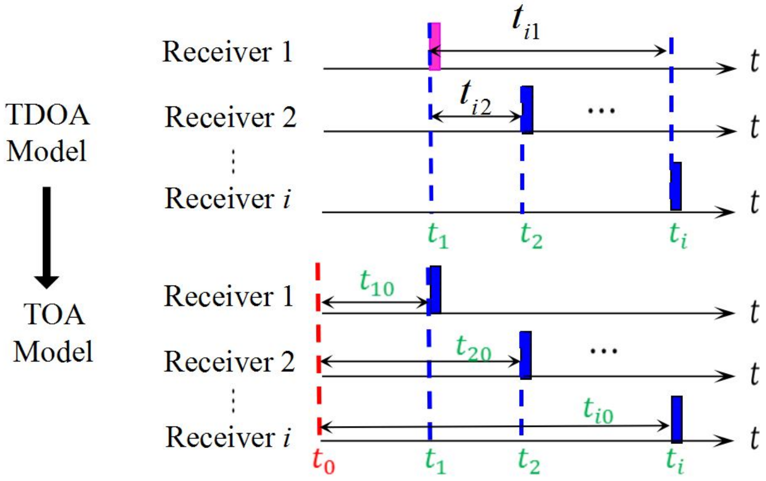


Figure 3.18: Transformation of TDOA model to TOA model.

mitigation in TDOA systems; second, like existing convex optimization schemes, it does not require *a priori* information about NLOS links or NLOS error statistics; and third, it achieves a much better performance than existing convex optimization schemes, which will be validated in both simulation and real experiments.

3.5.2 Proposed method for NLOS error mitigation in TDOA localization

Consider a TDOA system with M anchors, assumed to be receivers, where the first anchor is used as the reference, without loss of generality. The distance-difference between the source node, which is assumed to be a transmitter, to the i-th receiver and the reference receiver, is expressed as

$$\begin{aligned} r_{i1} &= r_i - r_1 = c\,(\Delta t_i - \Delta t_1) \\ &= (d_i + b_i - d_1 - b_1) + (n_i - n_1),\ i = 1, \cdots, M, \end{aligned} \tag{3.12}$$

where $r_i = c\Delta t_i$, Δt_i, and d_i are the measured distance, the signal propagation time, and the true distance from the source node to the i-th anchor, respectively, b_i

represents the NLOS-caused positive bias, and n_i is the range-measurement noise, which is commonly modeled as a Gaussian variable with zero mean and variance σ^2 [91].

The basic idea we develop here is to find an effective way to transform the TDOA architecture into a TOA architecture. In TOA systems, all anchors, assumed to be receivers in this paper, must be time-synchronized with the target transmitter; in TDOA systems, all anchors must be time-synchronized. In the TDOA model as shown in Fig. 3.18, if receiver 1 is designated as the reference, then t_{i1} will be the observed timing difference between the ith receiver and the reference receiver. In this case, since the receivers and the transmitter are not synchronized, the time the signal leaves the transmitter is unknown to all receivers. In the TOA model, as shown in Fig. 3.18, the time instant when the signal leaves the transmitter, t_0, is known to receivers, and t_{i0} is the observed traveling time of the signal from the transmitter to the ith receiver. Therefore, if t_0 is acquired in the TDOA model, a TDOA model can be transformed into a TOA model.

Since the t_0 is unknown in the TDOA model, we leave t_0 as a variable in the TDOA model and transform the TDOA model into a TOA model. The resulting TOA model with t_0 being a variable is described as

$$r_i = d_i + b_i + n_i \tag{3.13a}$$

$$(t_i - t_0)c = d_i + b_i + n_i, \ i = 1, \cdots, M. \tag{3.13b}$$

Squaring both sides of (3.13b) followed by some algebraic manipulations yields

$$\begin{aligned} c^2t_i^2+c^2t_0^2-2c^2t_it_0-d_i^2-b_i^2-2b_id_i &= 2n_i(b_i+d_i)+n_i^2 \\ &= \epsilon_i. \end{aligned} \tag{3.14}$$

Let

$$q_i = b_i^2 + 2b_id_i. \tag{3.15}$$

Eq. (3.14) simplifies to

$$\epsilon_i = c^2 t_i^2 + c^2 t_0^2 - 2c^2 t_i t_0 - d_i^2 - q_i. \tag{3.16}$$

Note that since the measurement noise n_i is generally much smaller than $b_i + d_i$ in practice, the n_i^2 term in ϵ_i can be neglected.

The nonlinear least-squares estimator of the unknown parameters $\boldsymbol{\theta}, \boldsymbol{q}, \boldsymbol{d}, t_0$ are expressed as

$$\begin{aligned} &\underset{\boldsymbol{d},\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{q}}{\operatorname{argmin}} \sum_{i=1}^{M} w_i \left(c^2 t_i^2 + c^2 t_0^2 - 2c^2 t_i t_0 - d_i^2 - q_i\right)^2, \\ &\text{s.t.} \quad q_i = b_i^2 + 2b_i d_i \end{aligned} \tag{3.17}$$

where w_i is a positive weight and $\boldsymbol{\theta}$ is the location of the target to be estimated.

Eq. (3.17) is non-linear and non-convex. Introduce two new variables:

$$s = t_0^2 \tag{3.18a}$$

$$h_i = d_i^2 = ||\boldsymbol{V}i - \boldsymbol{\theta}||_2^2, \tag{3.18b}$$

where s and t_0 in (3.18a) are treated as two variables. Eq. (3.18b) can be written in vector-matrix form by using the Schur Complement as [89]

$$h_i = \begin{pmatrix} \boldsymbol{V}_i \\ -1 \end{pmatrix}^T \begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{\theta} \\ \boldsymbol{\theta} & z \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_i \\ -1 \end{pmatrix}; \tag{3.19a}$$

$$\begin{pmatrix} \boldsymbol{I}_2 & \boldsymbol{\theta} \\ \boldsymbol{\theta} & z \end{pmatrix} \succcurlyeq \boldsymbol{0}, \tag{3.19b}$$

where z is a new variable added.

Since b_i and d_i are positive, q_i in (3.15) satisfies

$$q_i \geq 0, . \tag{3.20}$$

Eq. (3.17) is transformed into a semidefinite programming (SDP) problem as

$$\begin{aligned} &\underset{\boldsymbol{h},\boldsymbol{s},\boldsymbol{c},\boldsymbol{\theta},z,t_0}{\operatorname{argmin}} \sum\nolimits_{i=1}^{M} w_i \left(c^2 t_i^2 + c^2 s - 2c^2 t_i t_0 - h_i - q_i\right)^2 + \\ &\qquad \sum_{i=1}^{M} \rho(q_i^2 + s^2) \qquad (3.21) \\ &\text{s.t.} \qquad \text{Eqs. (3.19) and (3.20)} \end{aligned}$$

where ρ is a penalization factor, which is required when the problem is ill-posed [89]. Although (3.21) can be solved mathematically, its performance is not guaranteed.

Since t_0 is a key variable for the model transformation, we develop a few constraints on t_0 to improve the estimation accuracy of t_0. First, a geometric constraint:

$$(t_i - t_0)c + (t_j - t_0)c \geq ||(\boldsymbol{V}i - \boldsymbol{V}j)||_2, \quad i \neq j. \tag{3.22}$$

Since the ith and jth receivers as well as the transmitter can form a triangle, (3.22) holds because the sum of two sides of a triangle is always greater than or equal the third side. Second, since t_i is the arrival time of the signal at the ith receiver, we have

$$t_i \geq t_0. \tag{3.23}$$

Since the biases of NLOS links in (3.13b) are positive and typically much larger than the measurement noise n_i, we can write r_i as [89]

$$r_i = (t_i - t_0)c \geq d_i \quad \text{or} \qquad r_i^2 = (t_i - t_0)^2 c^2 \geq d_i^2. \tag{3.24}$$

From (3.12), we have

$$
\begin{aligned}
r_i^2 &= (t_i - t_0)^2 c^2 && (3.25a)\\
&= (t_i^2 - t_i t_0 + t_0^2 - t_i t_0) c^2 && (3.25b)\\
&\leq (t_i^2 - t_i t_0) c^2. && (3.25c)
\end{aligned}
$$

Eq. (3.25b) can be relaxed to be linear as Eq. (3.25c) since $t_i \geq t_0$ and the time instant are always positive; that is, $t_0 \geq 0,\ t_i \geq 0$. Therefore, we have $t_0^2 - t_i t_0 \leq 0$ and Eq. (3.25c).

From Eqs. (3.24) and (3.25), we have

$$
(t_i^2 - t_i t_0) c^2 \geq r_i^2 \geq d_i^2 = h_i. \tag{3.26}
$$

It is possible that (3.26) may not be feasible if the bias is $b_i = 0$ (i.e., no NLOS error) and the noise is negative (i.e., $n_i < 0$), since in such case, $r_i < d_i$, as shown in Eq. (3.13a). We resort to the soft-minimum method [8] to resolve this infeasible problem. In this method, a positive variable u_i is introduced in (3.26) to make it valid:

$$
\begin{aligned}
&(t_i^2 - t_i t_0) c^2 + u_i \geq d_i^2 = h_i, && (3.27a)\\
&u_i \geq 0. && (3.27b)
\end{aligned}
$$

A problem with this approach is that a large u_i will loosen this constraint. To ensure a strict constraint, we add the following item to the objective function

$$
\mu \sum_{i=1}^{M} u_i^2, \tag{3.28}
$$

where $\mu > 0$ is to be determined. With the soft-minimum method, the constraint in (3.27) tends to choose a proper value of u_i to ensure feasibility of the constraint while (3.28) in the objective function optimizes the value of u_i to make the constraint tight.

The SDP problem is summarized as

$$\begin{aligned} \underset{\boldsymbol{h},\boldsymbol{q},\boldsymbol{\theta},\boldsymbol{u},z,t_0,s}{\operatorname{argmin}} & \sum\nolimits_{i=1}^{M} w_i \left(c^2 t_i^2 + c^2 s - 2c^2 t_i t_0 - h_i - q_i\right)^2 + \\ & \sum_{i=1}^{M} \rho(q_i^2 + s^2) + \mu \sum_{i=1}^{M} u_i^2 \\ \text{s.t.} \quad & \text{Eqs. } (3.19), (3.20), (3.22), (3.23), (3.27). \end{aligned} \tag{3.29}$$

3.5.3 Simulation and experimental results

3.5.3.1 Simulation setup and results

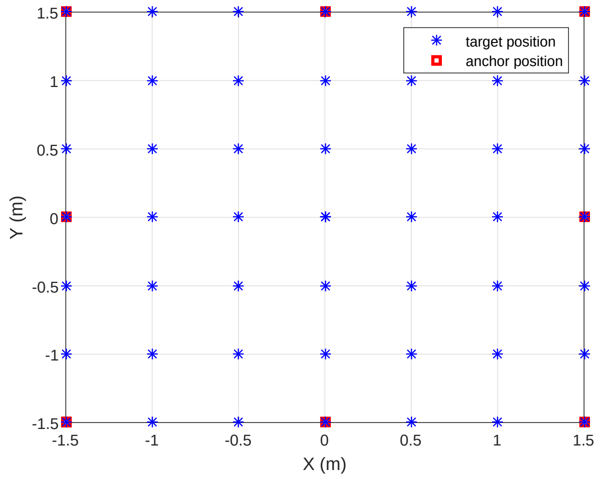


Figure 3.19: Simulation configuration.

The performances of the methods listed in Table (3.5) are compared via simulation and experimental data. The simulation configuration is shown in Fig. 3.19, which includes eight anchors and one source node. Forty nine locations of the source node are uniformly distributed in a $3m \times 3m$ space. The NLOS bias is assumed to be uniformly distributed over $[0m, 3m]$, and is unknown to all estimators. The mean-squared error (MSE) of the estimators is simulated in different noise conditions. The weighting elements w_i are all set to 1, the penalization factor is set as $\rho = 0.01$, and μ is set as $\mu = 1$.

Note that in the objective function in (3.29), timing information such as t_0 and t_i, and the square of the speed of light c^2 are directly involved in the calculation. Since c^2 is much greater than the difference between t_i and t_0, the calculation with computers involving these might cause a loss in precision since a tiny change in t_0 will cause large error in the final estimation. To avoid this potential precision loss, we multiply t_i by 10^8 while multiply c is scaled by 10^{-8}. This normalization does not affect the final result but can avoid the precision loss.

Table 3.5: Estimators considered in the performance comparison.

Estimator	Description
PROPOSED	The proposed estimator.
SDR-ROBUST	The robust semidefinite relaxation for NLOS mitigation in [91].
SDP	The SDP estimator in [94].
ML-LOS	The ML estimator with only LOS links in [89].

Simulation results are shown in Fig. (3.20), where

- Condition 1: Out of the 8 links, up to 5 are NLOS links;
- Condition 2: Out of the 8 links, up to 2 links are LOS links, which represents an unrealistically severe NLOS situation.

The performance of ML-LOS, which assumes the best scenario where all links are LOS links and ML estimator is used, is included as the benchmark. Note that the ML-LOS is not applicable for Condition 2 since at least 3 links are required for 2-dimensional positioning.

It is observed from Fig. (3.20) that under Condition 1, the proposed estimator significantly outperforms other convex optimization estimators, and its performance approaches that of ML-LOS when $\sigma \geq 0.3m$. In Condition 2, the unrealistically severe NLOS situation, the proposed estimator still outperforms other estimators when $\sigma \leq 0.4m$, which represents an unrealistically large range error (the maximum dimension of the space is $1.5m$); after this point, SDR-ROBUST starts to perform slightly better than the proposed method.

In addition, since the TDOA system has been transformed to be a TOA system, the convex hull problem, reference selection challenges, and difficulties due to the wider range of NLOS-caused ranging errors are completely avoided.

3.5.3.2 Experiment setup and results

A TDOA localization system with 80 MHz bandwidth Wi-Fi signal (IEEE 802.11ac) is set up in a laboratory to test the proposed algorithm. The setup is shown in Fig. 3.21. Due to limited number of hardware components available, only four receivers are used and all receivers are placed on the same wall (with the same x-coordinate). This is set up to determine the y-coordinate of the target because the receivers are placed on the same plane, and the receive antennas are separated by a maximum of $1.5m$ along the z-direction. If the x-coordinate of the target needs to be estimated as well, more receivers can be placed along the y-direction. An iPhone 6 plus is used as the transmitter. NLOS propagation for each target position is created by randomly blocking 0 to 4 links of the signal between the target and the receivers. For each position, 500 sets of data are acquired. The average error and its standard deviation (std) of the y-coordinate estimates are shown in Table (3.6). Because of the receiver gain is very low, due to limitation of available hardware, for some positions marked in red in Fig. 3.21, the received signals are too weak to generate valid results for any algorithm. Thus, these positions are excluded in the comparison. Also, since SDP-ROBUST requires prior information of the upper bound of NLOS bias, or the maximal NLOS bias, in this experiment, it is chosen to be 3 meters, which maximize its performance compared to other settings, like 2, 4 or 5 meters. These experimental results also show that the proposed scheme has a superior performance than state-of-the-art existing schemes.

Table 3.6: Average error and std for target positions in y-direction.

	Proposed	SDP	SDP-ROBUST
Average error (cm)	105.4	155.1	135.3
Error std (cm)	130.8	181	142.2

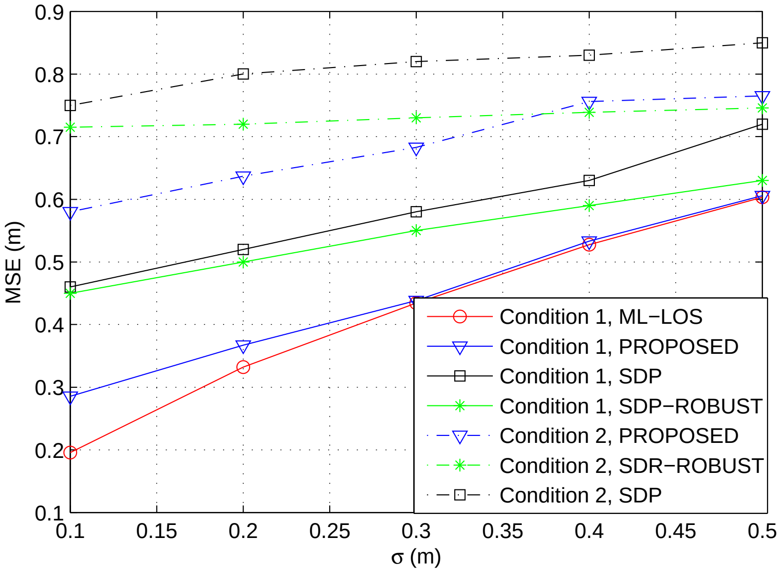


Figure 3.20: MSE results for different σ and NLOS conditions.

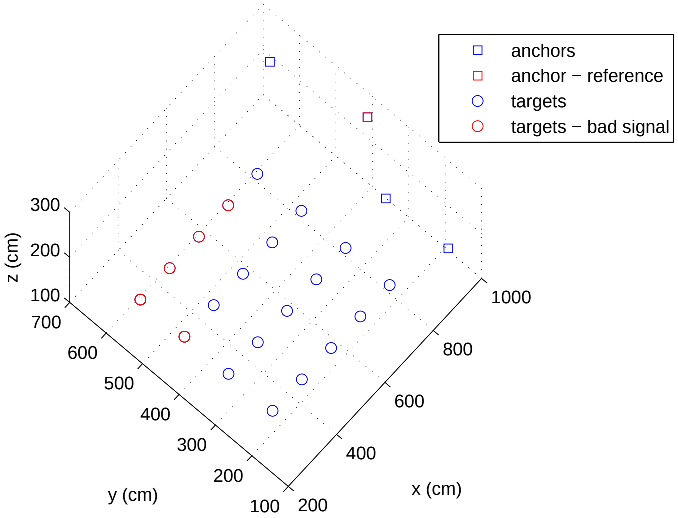


Figure 3.21: Experiment setup.

Chapter 4: Image-tag-based indoor localization with deep learning

4.1 Introduction and motivation

In previous chapters, RF signal-based indoor localization with trilateration methods such as the TOA and TDOA, are discussed. However, in those methods, the accuracy of distance related information is easily degraded, due to many RF signal propagation related problems, including multipath problem, the NLOS problem and the synchronization problem. In order to conquer those problems, several algorithms have been proposed. Although the proposed algorithms are able to mitigate the errors, it's still difficult to completely eliminate such errors. In this chapter, we resort to some other methods to obtain the distance and angle information, like the image based method, so that those issues caused by the RF signal propagation are completely avoided.

In fact, image-based methods such as the image-based indoor localization have been existing for a while. The basic idea of image based localization is to localize the camera through matching the cemera's captured image to one of the images in the database. The images in the database are previously captured within the building and each is tagged with the location where they were captured. Therefore, if the newly captured image can be matched to one image in the database, the location of the camera can be estimated.

However, there are several common issues and drawbacks for the image matching or feature matching method.

- One issue is that doing localization based on naturally-occurring features can be computationally intensive, but we want to be able to run our application on a mobile device with limited computing power.
- Another issue is that, in a large building, there can be many locations that

have a similar appearance, thus potentially confusing an image-based recognition system.

- The third issue is that it's unreliable to the environmental changes since all algorithms are based on natural features.
- The fourth is the heavy calculation required in the image or feature matching system.
- The last but not the least issue is that it's not scalable. If we want to extend the coverage of an existing image based localization system, we need to train the whole system again with the additional images.

The matching method in image based localization is not a trilateration related method, as it doesn't include the distance estimation. In this chapter, we propose an image-tag-based localization, which tries to mimic the RF signal-based indoor localization with trilateration to estimate the distance and angle related information from the image. And then we use such information to estimate the target's location.

Compared to the RF signal-based indoor localization method, the signal propagation related issues, such as multipath and NLOS effects, are completely avoided in the distance estimation with proposed image-tag-based indoor localization.

Compared to existing image-based indoor localization, the proposed image-tag-based localization has several advantages. In this method, the tags only need to be trained for time, and then they can be used anywhere. The system is easy to set up and simple in application. Since the tags can be distributed anywhere, there are no blind spot problems and the accuracy can be guaranteed. Also, all information is extracted from the segmented tags, and it has nothing to do with the environment, and therefore it's robust to environment change. Finally, if we have enough trained tags, it's easy to extend an existing system by simply distributing additional tags to those extended parts without any additional training.

4.2 Related works

Image-based indoor localization has drawn lots of attention [106]. Different kinds of methods have been proposed, like image matching using SIFT features [4, 107, 108]. The basic idea behind is that a database of structured images of the indoor environment is constructed by using image matching and the bundle adjustment algorithm. Then, each image's relative pose, like its position and orientation, is estimated, and the semantic locations of images are tagged. The location of the user can then be determined by comparing a photo taken by the mobile phone to the database. This is done by combining quick image searching, matching, and relative orientation. Although the searching is speeded up, constructing the visual words is complex and costly. In other methods [109, 110], a topological map is built from a series of images, a panoramic photo, or a video sequence, and then is refined by learning vector quantization (LVQ). During the searching stage, the nearest neighbor rule is used to detect the similar regions in the query image. This method assumes that the navigation path is unique in the topological map. Therefore, the query image can be misclassified in some cases.

However, there are several issues remaining in the image-based localization method. The first is the blind spot issue. Since many parts of the building look similar, algorithms can't distinguish them and therefore will fail to match the location correctly. The second issue is scalability. For example, if a new part of a building needs to join the originally existing positioning system, the images for the new parts need to be re-trained together with the old images in the database to extract the features. Also, both feature extraction and matching processes require heavy calculations. Besides, it's not robust to environment changes. For example, if the decoration of part of a building changes, the photos taken from that part will not be able to match any images in the database. The fifth is the high requirements for hardware, as all photos for each building should be pre-stored in some server or database for further use. The sixth but not the least problematic issue is the repeated heavy calculation tasks, as each building needs to repeat the whole process (taking all photos, putting them in a database and extracting the features).

In addition, camera pose estimation is also used in the indoor localization [111,116]. However, the accuracy of pose estimation relies on the accuracy of local corner or point detection in the image. When the image is blurred or taken from far away, the accuracy of corner/point detection will decrease and, therefore, the accuracy of pose estimation decreases, which can be seen in the results of [111].

4.3 Image tag-based localization with deep learning

The whole pipeline of the proposed image tag based localization method is shown in Fig. 4.1. At first, N different tags are designed, where N should be large enough to cover different areas. Then three convolutional neural networks (CNN) are trained to predict the tag ID, distance, and angle, respectively. Note that these tasks are only necessary for one time, which means that once the tags are designed and the CNN models are trained, it can be used anywhere without additional training. Then, to build a localization system for a building, the tags are distributed within the building–usually attached on the walls. When doing localization, as shown in Fig. 4.2, a user in the building can randomly take a photo. If there are one or more tags in the photo, they are segmented out from the photo and then used to predict the tag ID, the distance, and the angle information with the previously trained CNNs. The distance here refers to the distance between user (or camera) and the tag, while the angle refers to the angle formed by the line perpendicular to the tag and the line between the user and the tag, shown as d and θ, respectively, in Fig. 4.2. Finally the location of the user can be estimated with the method shown in Fig. 4.3, where an intersection between a semi-circle and a line could be found and it's the location of the user. If there are more than one tag in the photo, the accuracy will improve since more information is available.

In the following subsections, each part of the proposed flow will be introduced in order.

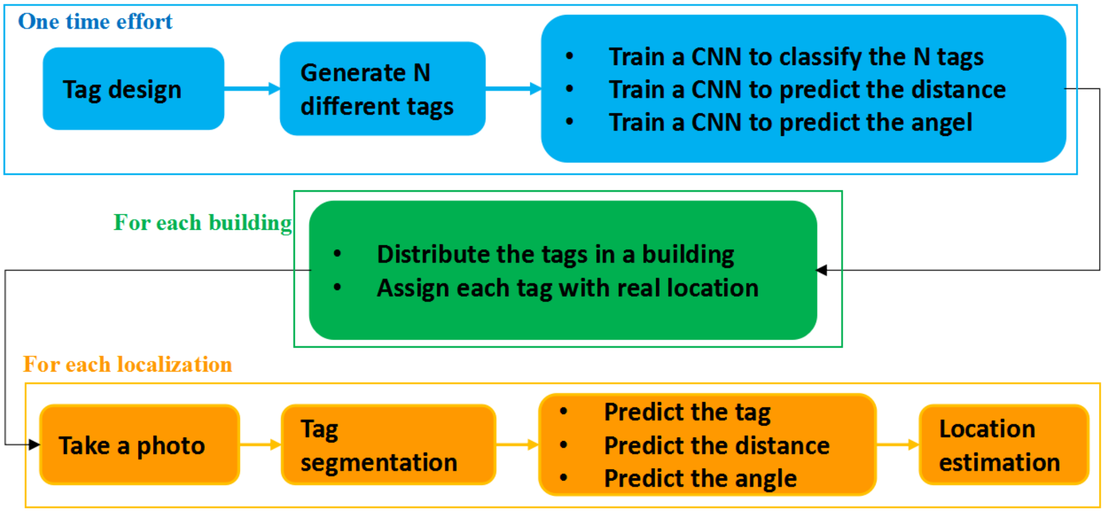


Figure 4.1: The general flow for image tag based localization.

4.3.1 Tag design

As discussed before, the tag is used for information extraction through CNN. To fit this idea, the design of the tag should meet the following requirements: 1) It's easy to create a large amount of different tags in order to cover a large area; 2) It has simple features, so that it can be seen clearly from far distance; 3) It has non-symmetric features, as the positive and negative angles need to be distinguished; 4) For reliable segmentation, the features should be shape-based, not color based. Although it's easier to segment the tag from the photo with color-based features, which are used in [111], it's not robust as the change of the lighting in the building may cause the segmentation to fail; 5) It should have a common feature for all tags for the purpose of tag segmentation, distance, and angle information extraction.

There are existing tags designed for different purposes. A list of current detectors is quoted [119]: each of them uses a kind of visual tag with different main characteristics. A few examples are shown in Fig. 4.4.

QR-Code: It uses a quadrilateral visual tag where the information is stored in a binary pixel matrix from 1817 to 7089 characters, depending on the alphabet

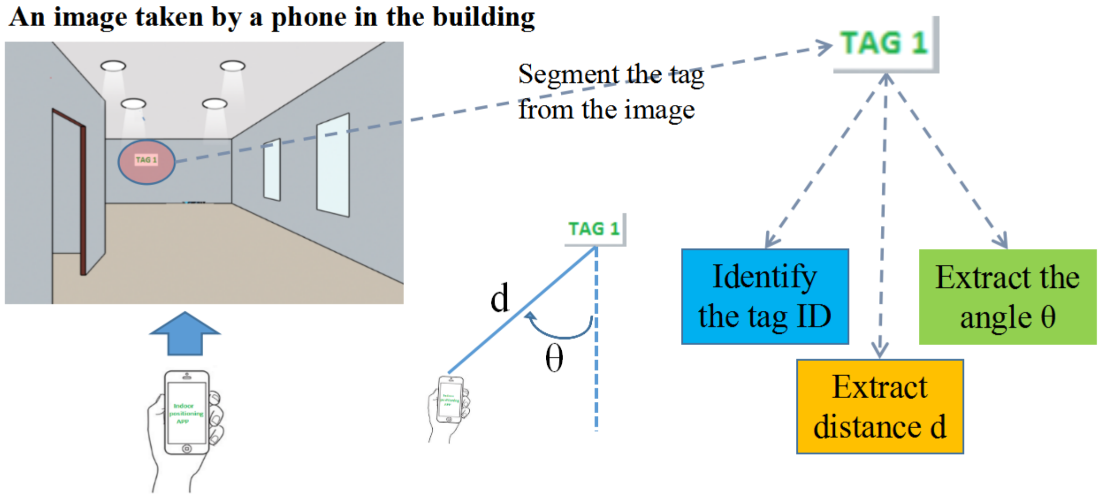


Figure 4.2: The general flow for image tag based localization.

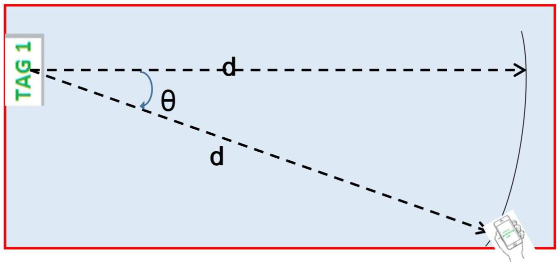
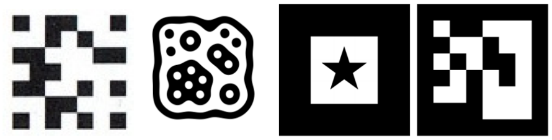


Figure 4.3: Precisely locate the target with extracted information.

used. The main problem of its visual tag is its detection speed, which is rather slow, due to the significant amount of information stored inside the visual tag.

Maxicode: This system, which is used in the postal service of the USA, is a quadrilateral visual tag, where the information is stored in a hexagonal binary pixel matrix. Unlike the previous one, its visual tag can store up to 93 characters, which means it can be quickly detected, since the information stored is significantly less in comparison with the previous one.

CyberCode: In this system, the information is stored in a binary pixel matrix. Unlike the previous cases, its visual tag is designed to be detected quickly and



(a) Cybercode (b) ReacTIVision (c) ARToolkit (d) ARTag

Figure 4.4: Existing tags designed for different purposes [119].

robustly in front of significant perspective views.

ReacTIVision: This system, which is used in the famous Reactable, uses a very simplistic quadrilateral visual tag, wherein the information is stored according to the shape of its internal pattern.

ARToolkit: It's considered the first in positioning applications and uses a quadrilateral visual tag where the information is stored according to the shape of its internal pattern. Moreover, the fact of using a detection system based on the recognition of the pattern of its visual tag, pro-vokes that this pattern should be at once complex, in terms of being unusual to find it in a natural environment, and simple, in terms of being able to be quickly detected.

ARTag: This system, successor of the AR-ToolKit, uses a quadrilateral visual tag, wherein the information is stored in a binary pixel matrix, which, in this case, is protected by an FEC code. As in several previous cases, its visual tag is quick to detect, due to the little information stored in the matrix. Furthermore, it is robust against rotation and perspective views. In specific terms, its visual tag can store up to 36 characters.

As discussed above, the QR-Code and Maxicode are only used for near distance applications. While the other tags could be read from long distances, they either miss a common feature or a non-symmetric feature. Motivated by the tags shown in Fig. 4.4, we designed a vision tag that meets all requirements.

The designed tag is shown in Fig. 4.5. The upper left, non-symmetric square feature is composed of small circles, which are designed for segmentation, distance, and angle information extraction. As the square forms the boundary of the tag, the segmentation is done by identifying the boundary. The reason for the use of

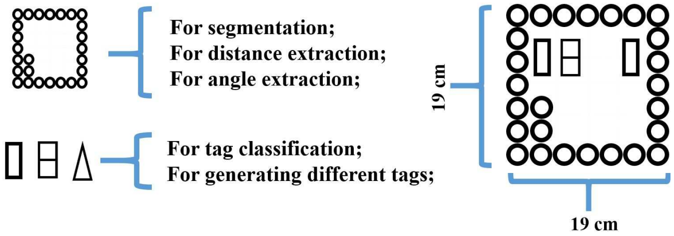


Figure 4.5: Tag design.

small circles to form a square instead of using a square with solid lines is that the small circles enrich the features. The rich features can help distinguish the tag from the background and, therefore, simplify segmentation. However, overly complex features will cause issues; for example, it can't be seen clearly from a far distance. So it's a trade-off between the simple features and the visibility from a far distance. Also, this square shape is the common part of all tags and its shape and size will change when seen from different distances and angles. So the distance and angle information is encoded in the distortion of the square.

The bottom left shows some example features that are used to fill the inner blank space of the square to generate different tags through combination and permutation of their features. Note that more such features can be designed and, therefore, a large number of different tags can be designed easily. The right side of the figure shows an example tag with filled features.

4.3.2 Tag segmentation

Tag segmentation is critical, since all the following processes are based on correct segmentation. Different kinds of approaches have been proposed to do image segmentation and achieve a good performance, like the method using Deep Convolutional Encoder-Decoder Architecture [112], which aims at precisely segmenting the boundary of an object, like a dog, from an image. Instead of precisely segmenting the exact boundary of the square feature object, we only need to use a square window to segment the square feature (note that the outer square feature could become a parallel, echelon, or other shape when viewed from different angles and distance). And we want to make full use of this identical common feature for all tags. The square, as shown in Fig. 4.5, was used to develop a simple, accurate, and robust segmentation method. Another consideration is that there may be more than one tag in a photo, and therefore each tag should be correctly segmented. A good segmentation should be correct, complete, and precise, as shown in Fig. 4.6.

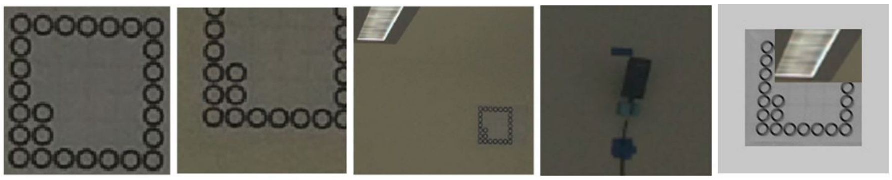

(a) Good **(b) Not complete** **(c) Not precise** **(d) Not correct** **(d) Occluded**

Figure 4.6: Example of segmented tags.

The best known template matching methods are SIFT [117] and SURF [118]. However, SIFT is not robust to lighting change and blurring, and it will include the incomplete and occluded segmentations, which are not desired in the proposed method. To correctly find a template in the target image, at least three key points are required. In addition, additional operations are required to segment multiple tags in an image. To overcome these issues, we develope an accurate and robust segmentation method for possible multiple tags with three steps: key feature searching, clustering, and prediction.

Key feature searching learns the key features in the template and then tries to find out the similar features in the target image and locate them. As shown in the top of Fig. 4.7, many key features are found (the dots in the top-right plot); among them, part of the key features are caused by the environment instead of the tags. In the second step, the hierarchical clustering method is employed to cluster these dots [114]. With some proper threshold, several main clusters can be found. As the example shown in the middle of Fig. 5, there are three main clusters, which are labeled as c1, c2, and c3, respectively. Based on each cluster of dots, we segment the patches out of the image with a square window that can exactly cover all key points. The segmented patches could be complete tags, incomplete tags, occluded tags, or non-tags, as shown in Fig. 4.6. In the third step, a simple CNN is trained to judge whether the segmented patch is a complete tag or not and it will keep only the patches with complete tags and discard all the other patches.

To be specific, the two-class CNN mentioned here is basically a one class-CNN problem, as it only tries to predict whether the input is a tag or not [120]. The issue for one-class classification is that it's easy to define what the class is. For example, the input segmented patches are judged to be tags as long as they contain a tag. However, it's difficult to define what's not a tag, as the environment can be different and it's therefore not able to emulate all scenarios that are not tags.

However, in our case, the not complete and occluded tags are important examples of the non-tag category. In the training process, lots of different kinds of not complete and occluded examples are used to tell the CNN that these are non-tags instead of tags.

4.3.3 Tag classification

Since tags are distributed in the building and associated with real positions, if the segmented tags can be correctly identified, hence the general location of the camera could be known. Since each tag has its own special features (or a combination of features), an exact mapping between tags and ID can be built. To identify each tag, neural network with manually made features are used in [111]. Those manually made features are specially designed for their tags and are not guaranteed to be the best features. To improve the performance, a CNN, which is able to extract the required features, is trained to predict the ID of the segmented tag. The configuration for the classification-based CNN is shown on the left side of Fig. 4.8.

4.3.4 Distance and angle extraction

As discussed in previous sections, the distance and angle information are to be extracted from the segmented tag, as shown in Fig. 4.2.

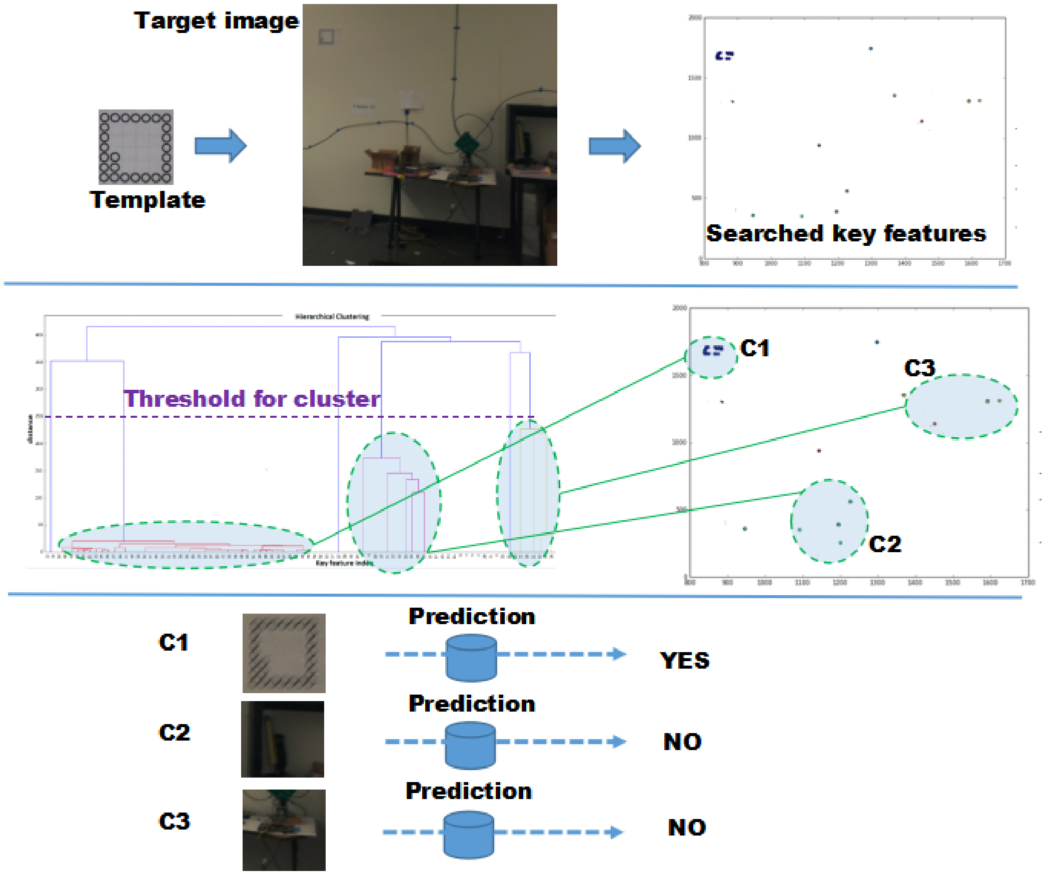


Figure 4.7: Tag segmentation.

The distance refers to the distance between the tag and the camera. It's understandable that there is a certain mapping relationship between the distance and the shape and largeness of the tag that appears in the photo, even when the photo is taken from different angles. To be specific, the mapping between the tags and distances is a many-to-one function mapping; mathematically, it's defined as

$$d = f(\text{tag}_d), \tag{4.1a}$$

$$d \neq f(\text{tag}_{d'}), d' \neq d, \tag{4.1b}$$

Classification based CNN	Regression based CNN
input (298*298*1 gray image)	input (298*298*2 gray image)
Average pooling (2,2)	Average pooling (2,2)
Convolution-2D-32	Convolution-2D-32
Activation-relu	Activation-relu
Convolution-2D-32	Convolution-2D-32
Activation-relu	Activation-relu
Max pooling (2,2)	Max pooling (2,2)
Dropout (0.25)	Dropout (0.25)
Convolution-2D-64	Convolution-2D-64
Activation-relu	Activation-relu
Convolution-2D-64	Convolution-2D-64
Activation-relu	Activation-relu
Max pooling (2,2)	Max pooling (2,2)
Dropout (0.25)	Dropout (0.25)
Convolution-2D-128	Convolution-2D-128
Activation-relu	Activation-relu
Convolution-2D-128	Convolution-2D-128
Activation-relu	Activation-relu
Max pooling (2,2)	Max pooling (2,2)
Dropout (0.25)	Dropout (0.25)
Convolution-2D-256	Convolution-2D-256
Activation-relu	Activation-relu
Max pooling (2,2)	Max pooling (2,2)
Dropout (0.25)	Dropout (0.25)
Full connection (4096)	Full connection (4096)
Activation-relu	Activation-relu
Full connection (1000)	Full connection (1000)
Activation-relu	Activation-relu
Full connection (30)	Full connection (1)
Soft-max	linear
Output (M classes)	Output (1)

Figure 4.8: Configuration for the classification and regression based CNN.

where d is the real distance between the tag and the camera and tag_d represents the tags taken at distance d. Note that d is not related to the angles, θ, as shown in Fig. 4.3.

An example of such mapping is shown in Fig. 4.9, where the top tags are taken from 3 meters away while the bottom ones are taken from 7 meters away at different angles.

Therefore, theoretically, a function, as Eq. (4.1) exists that can find the distance through each tag. Unfortunately, such a function is difficult to build mathematically due to the complexity of the changing in the shape of the tag in the image. A good way is to approximate this function through neural networks, like CNN, which is a good candidate, as it's able to extract the required features automatically. Since the distance is a continuous value, the regression-based CNN could be used.

Based on the above analysis, if enough and correct data and labels are provided, the CNN should be able to approximate the function in Eq. (4.1) with infinitely small error.

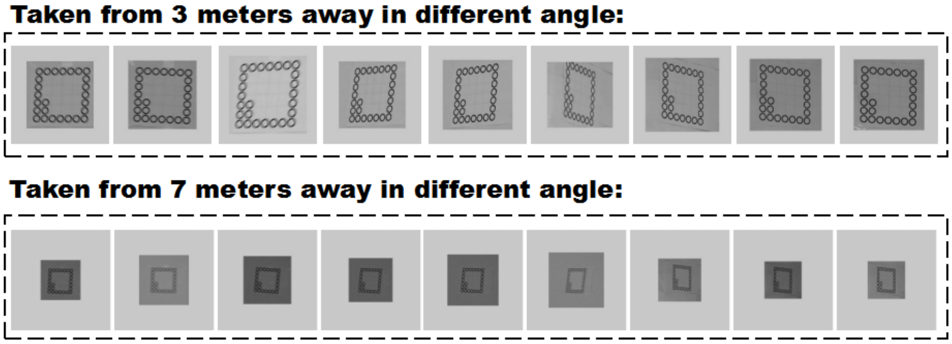


Figure 4.9: An example for the mapping relationship between distance and the size of the segmented tag.

For the angle information, some mapping relation between the angle (θ) and the shape of the tag in the photo can also be observed, which can be seen in Fig. 4.9. However, the shape of the tag in the image is not only affected by the angle, θ, it's also affected by the rotation angle of the phone, β, as shown in Fig. 4.10.

So, the mapping between the angle, θ, and the shape of the tag and rotation angle, β, is a function mapping that can be mathimatically described as

$$\theta = f(\mathrm{tag}_{\theta}, \beta), \tag{4.2a}$$

$$\theta \neq f(\mathrm{tag}_{\theta'}, \beta'), \theta \neq \theta', \beta \neq \beta', \tag{4.2b}$$

where tag_{θ} represents the tag taken at the angle of θ.

One issue for this mapping is that the rotation angle, β, is difficult to obtain. However, when the camera is rotated in different angles, the tag will locate in

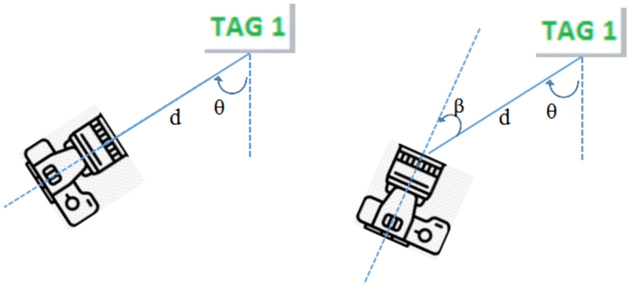


Figure 4.10: Camera rotation. Left: without rotation; Right: with rotation β.

different locations in the image, as shown in Fig. 4.11. Also, the mapping between the rotation angle, β, and the location of the tag in the image is a one-to-one function mapping:

$$\beta = f(\mathrm{tag}_l), \tag{4.3}$$

where the tag_l represents the location of tag in the image. Therefore, we have:

$$\theta = f(\mathrm{tag}_\theta, \mathrm{tag}_l), \tag{4.4a}$$

$$\theta \neq f(\mathrm{tag}_{\theta'}, \mathrm{tag}_{l'}), \theta \neq \theta^{'}, \mathrm{tag}_l \neq \mathrm{tag}_{l'}. \tag{4.4b}$$

Based on the above analysis, the mapping between the tag and the location of the tag in the image and the angle, θ, is a function mapping. Similar to the distance extraction, instead of building an exact function mathematically, we try to approximate it with a CNN.

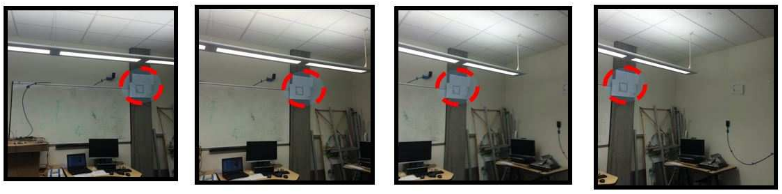


Figure 4.11: An example for rotating the phone horizontally when fixing the location of the phone and the tag on the wall.

To clearly tell the CNN the location of the tag in the image, we manually make an image, as shown in the left part of Fig. 4.12, where some black points in a blank (white color) image are used to represent the location of the tag in an image. The size of the manually-made image is shrunk to the same size of the segmented tag, so that they can be concatenated together and passed to CNN. Since we have already segmented the tag from the photo in previous sections, we have already known the location of the tag in the photo and therefore, this manually-made image could be easily done.

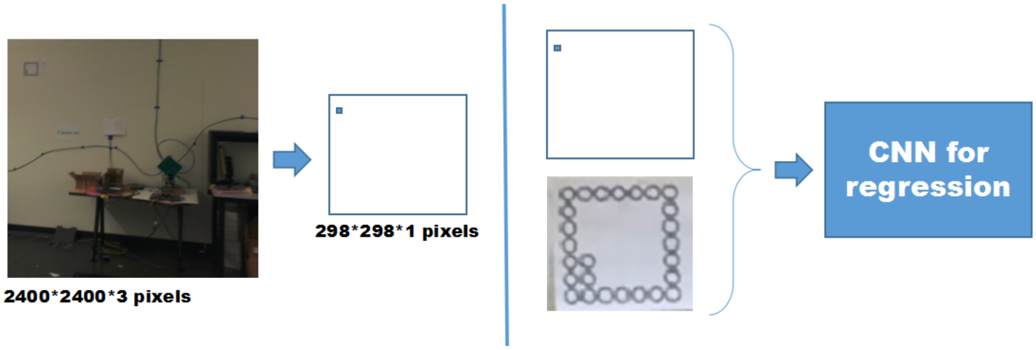


Figure 4.12: Manually-made image for CNN. Left: An example for a manually-made, black-and-white image to represent the location of the tag in the photo. Right: Concatenate the manually-made image and the segmented tag as the input to CNN.

4.3.5 Location estimation

With the tag ID, distance, and angle information, the location of the camera could be estimated using the method shown in Fig. 4.3. Note that, if more than one tag appears in a photo, the accuracy of location estimation could be improved.

4.4 Experiments and results

To validate these ideas and methods, we conducted experiments and tests in our lab according to the flow in Fig. 4.1.

The tags with a size of $19cm \times 19cm$ were printed with black-and-white color, as shown in Fig. 4.5. To test the tag segmentation and tag classification, only 10 different tags were used, due to the time limits. In the testing, each tag was attached on the wall in the lab and around 1000 photos were taken for each tag from different distances and angles with an iPhone 5S. To speed up the photo taking process, the burst function in iPhone is used. However, there were many blurred images with the burst function.

In the segmentation, since the tag in the photo could be small or large and have different shapes, a square shape was used to segment the tag from the photo. As the segmented images vary in size, they were padded with some value (e.g., 128), to the same dimension (e.g., 198×198 pixels), as shown in the bottom of Fig. 4.9, where the small dark square is the segmented image, while the outer gray square is padded with some value.

In the data set, there are around 10,000 photos for 10 tags. After discarding about 500 bad images that couldn't be recognized by average human sight capability, all photos were segmented using the proposed methods. There were about 9,100 photos that could be correctly segmented, which is about 96% of the total photos. For the failed cases, many of them were blurred photos.

After the segmentation, each of the segmented tags is labeled with its own ID, or the label for CNN. The classification-based CNN, as shown in the left part of Fig. 4.8, is trained with 80% of the segmented tags and validated on the remaining 20%. The validation accuracy is around 98%. This accuracy could be improved further with larger training sets, as less than 1,000 training data per class is a small data set compared to the Cifar10 and ImageNet datasets.

To train the regression-based CNN for distance and angle prediction, an experiment has was set up in our lab, as shown in Fig. 4.13, where there are 117 training positions and 46 testing positions. Due to the space limitation, the maximal distance was around 8 meters. For each training and testing position, around 300 photos were taken with the burst function in iPhone 5S. Therefore, there were around 30,000 photos for training and 10,000 photos for testing.

In the training, each tag in the photo was segmented and labeled with its real

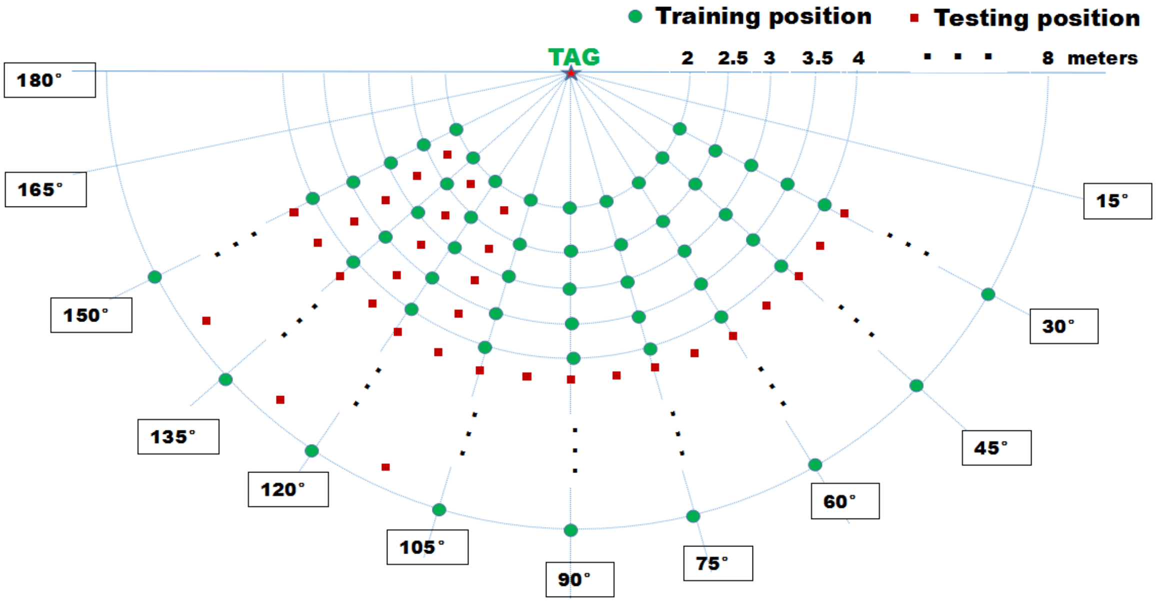


Figure 4.13: The setup for distance and angle extraction training and testing.

distance and angle. Note that the distance and angle were normalized to within the range of $[0, 1]$ to facilitate the learning. For the distance, it was normalized by 10 meters; for example the original 5 meters in distance was normalized as 0.5. And the angle was normalized by 180° (i.e., the 90° was normalized as 0.5).

All data for training (here refers to the segmented tags) were normalized with general mean and variance calculated from all training data. While in validation, all testing data were normalized with the mean and variance from training data.

Both distance and angle were trained with the same regression-based CNN architecture, as shown in the right side of Fig. 4.8, and with the manually-made image, as shown in Fig. 4.12. The segmented tag and the manually-made image are both grey images with only one channel. Note that the distance and angle are trained separately. In addition, to simplify the experiments, we only consider the two-dimensional case, or we only consider the x-axis and y-axis and discard the z-axis by putting the tag and the phone on the same height.

The testing results for distance and angle are shown in Fig. 4.14 and 4.15,

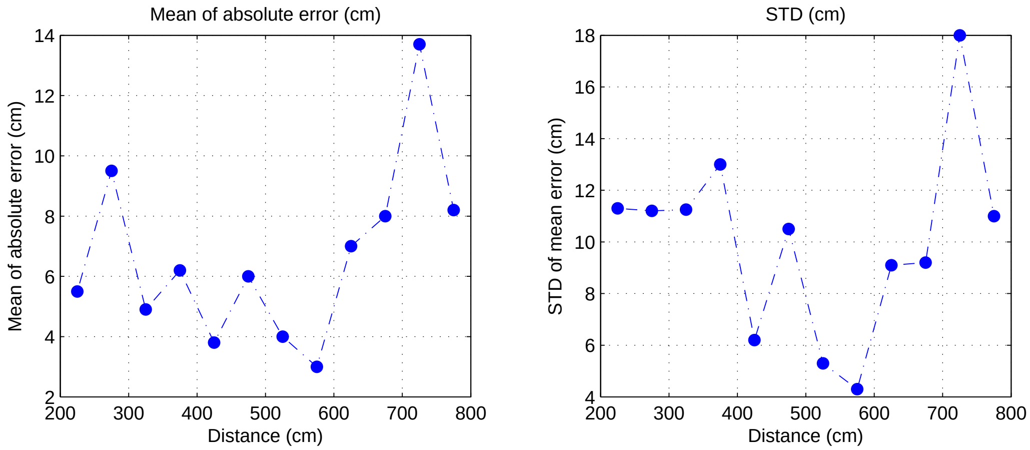


Figure 4.14: The testing results for distance extraction.

respectively.

The distance prediction, as shown in Fig. 4.14, has a low mean of absolute error, most of which is less than 10 centimeters (cm). It also has a low standard deviation (std).

The mean of absolute error for angle prediction, as shown in Fig. 4.15, is less than 6°, with half of them less than 3°. The std of angle error is around 0° to 6°, with a burst of error of 12° when the angle is 90°. In general, the error of angle increases when the camera is around 90°, probably because in these areas, the change of the shape in the tag is small and difficult to recognize and distinguish by the CNN.

To test the localization estimation, the experiments with one or two tags were set up in the lab.

In one tag scenario, each photo contain one tag. The position of the phone was found through the intersection of the semi-circle and the line, as shown in Fig. 4.3. The configuration and the results for one tag location testing is shown in Fig. 4.16, where the dots represent the testing locations. For each testing location, around 100 photo were taken. From the testing results, which are shown as the red triangles in the figure, the average location error is around 1 meter. However,

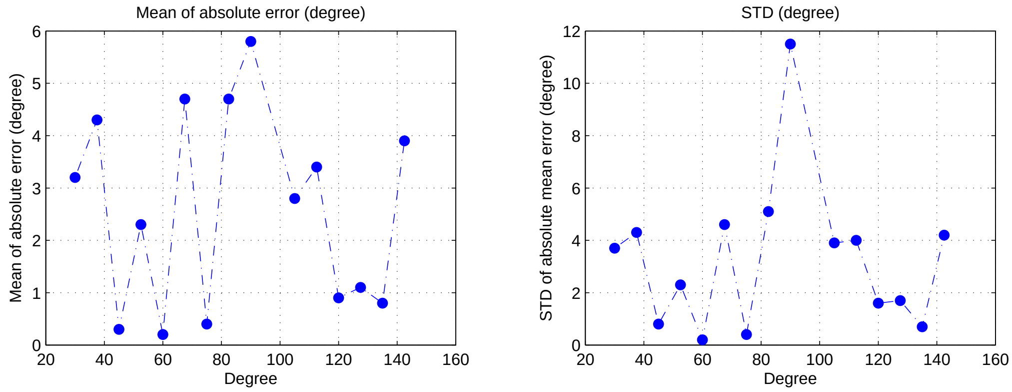


Figure 4.15: The testing results for angle extraction.

the location error around the 90° direction could be up to nearly 2 meters, as the angle prediction in this area has a relatively higher error. Note that the location error will also increase with the distance, as the error of angle will be enlarged along with the distance.

In two tag scenarios, each photo taken contains two tags. So, for each testing position, two distance and two angle information could be used to estimate the location. In the estimation, the position is found through the intersections of two circles and the angle information is used to distinguish which intersection is the one we are looking for, as there are two intersections for two intersected circles. Due to the high accuracy and robustness of the distance prediction, the average location error is less than 20cm.

4.5 Performance analysis

As discussed in previous sections, both distance, d, and the angle, θ, have their own function mapping. In theory, it's possible to approximate the function with small enough errors. The possible causes for the errors in the results in the last section, especially for the error of the angle are discussed below.

Firstly, the training data is not enough. Both distance and angle are continuous value. However, for the distance, only the data that locates on the limit locations

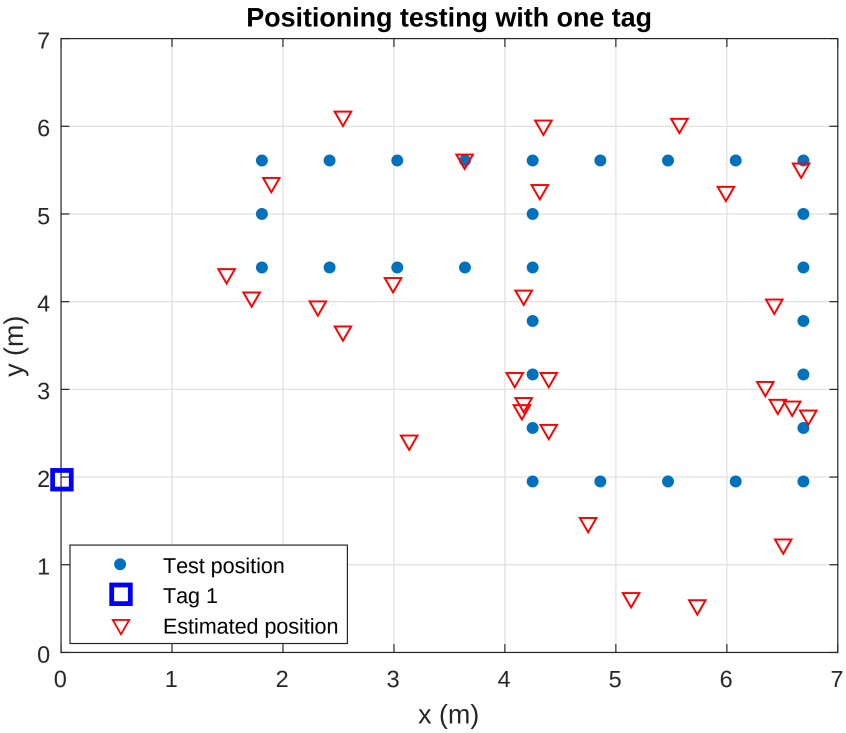


Figure 4.16: The location estimation results for only one tag.

are given, like 2m, 2.5m, 3m, etc. For the data about angle, 15° is used to sample the data from 30° to 150°. The potential method to improve the performance is to increase the sampling locations for both distance and angle (e.g., the 5° could be used to sample the angle data).

Secondly, as the angle is related to the shape instead of the largeness of the tag, it's possible to up-sample the tag when the tag is small or far away, to increase its resolution and therefore to help the CNN perform better.

Thirdly, it would be helpful to increase the measurement accuracy of the data with such information the real distance or angle of the sampled data, as the errors in Fig. (4.14) and (4.15) are not consistent, especially for the error caused by the angle (e.g., the mean of absolute error for the angle from 40° to 80° jumps between 0° to around 4°, which is more likely caused by measurement errors).

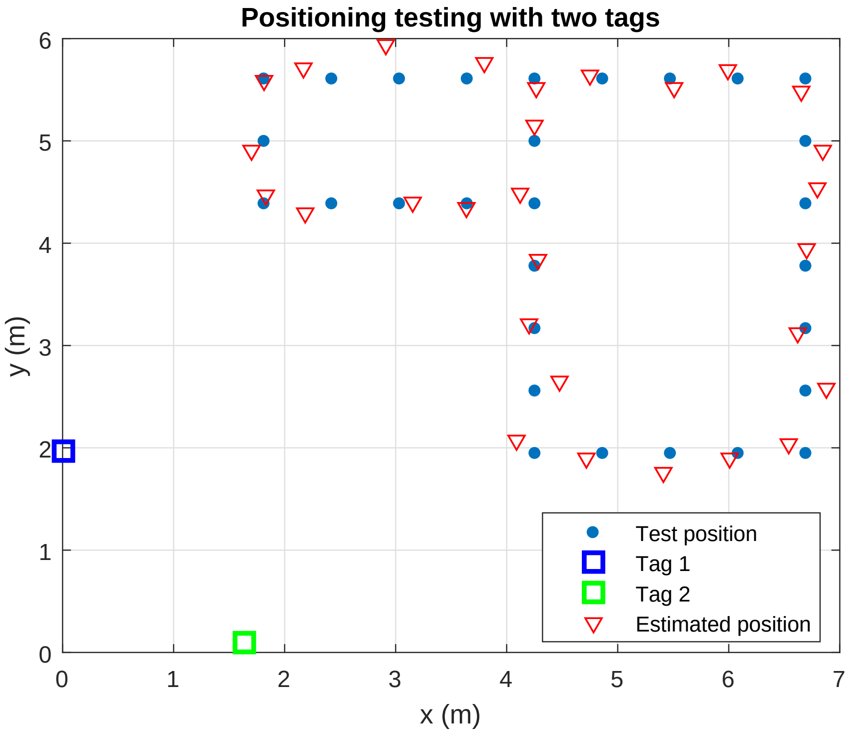


Figure 4.17: The location estimation results for two tags.

4.6 Conclusion and discussion

In this chapter, an image tag-based method for indoor localization is proposed. It trains several CNNs to classify the tag ID and to predict the distance and angle information of the segmented tag in the photo taken by the camera. With this information, the location of the user could be exactly identified.

It has several advantages over existing image-based localization methods. Firstly, it doesn't have the blind spot issue; Secondly, it only needs to be trained one time and then can be used anywhere; Thirdly, it's scalable; Fourthly, the calculation requirements for the predictions are not high compared to those of image matching in image-based localization. Fifthly, it's robust to environmental changes. In addition, the localization accuracy could be high if the tags are densely distributed.

These ideas are validated by several experiments set up in the lab. The localization experiments show that with one tag only, the average localization accuracy is around 1 meter. If two tags are used, the average localization accuracy could be less than 20cm.

Consequently, the experimental results prove that the proposed method is practical and provides good performance.

However, there are still several issues needs to be resolved in the future. The accuracy of the angle should be improved further, as the localization error caused by the angle will increase with distance. Also, there may be cases that no tags are taken in the photo if the tags are not so densely distributed in a building. A possibly practical application is that it can be combined with the inertial system in the phone to achieve an accurate and simple localization system [115]. The advantage of inertial sensor-based localization is that it can track the phone from a known position without any additional information. But it needs a known position from the start, and the degree of error will increase and accumulate as the track lengthens. Since the proposed system provides accurate positioning in a two-tag case, these accurate positions can serve as the starting known positions for the inertial sensors to track the phone.

Chapter 5: Conclusion and future works

5.1 Conclusion

This dissertation proposes solutions to address the technical challenges associated with indoor localization using trilateration, which uses geometric distance to obtain the location of the user. In RF signal-based indoor localization, TOA and TDOA are widely used to determine the distance related information regarding the transmitter and receiver. However, several issues remain unsolved, like the synchronization and NLOS problems, both of which are caused by the signal propagation nature of RF.

The first part of this dissertation focuses on resolving the issues of synchronization and NLOS. For the synchronization problem, we propose an algorithm to resolve the synchronization problem in the high speed multi-channel data acquisition unit in a FPGA. Armed with this algorithm, the sampling unit can sample four channel of input signal at the speed of 3 Gsps while they are synchronized. Also, we propose a synchronization-free positioning network architecture that eliminates the need of timing synchronization.

For the NLOS problem, which greatly degrades the positioning performance, we propose a SDP with soft-minimal method to eliminate the infeasibility problem. To further mitigate the NLOS error, an NLOS link identification method with bias deduction to mitigate the NLOS error in TOA systems. For TDOA systems, NLOS mitigation is more difficult, since a reference should be fixed first. To overcome this problem, we propose a method to transform the TDOA architecture into a TOA one, and then form an SDP problem with new constraints. The proposed methods outperform the existing algorithms in both simulation and experiments.

To obtain the distance information, the RF-based techniques, like the TOA and TDOA, encountered several difficulties, as discussed above. We propose an image tag-based technique to obtain the distance and angle information, which can completely avoid such issues that are common in RF-based localization and can increase the positioning accuracy. In the proposed method, after the segmentation of the tags from the image, information such as the tag ID, distance, and angle with reference to the camera are retrieved through deep neural networks. The camera position is finally reliably and accurately estimated from such retrieved information.

5.2 Future works

In this dissertation, the indoor localization with trilateration is discussed with RF signal and image processing, respectively. For RF signal based indoor localization, solutions to synchronization and NLOS problems are proposed. However, several issues/problems still need to be addressed: 1) In severe NLOS condition, it's still difficult to improve the localization accuracy. 2) In the synchronization free model, since the repeater is introduced, which complicates the whole system, it's necessary to discern how to simplify the system.

For the image-tag-based localization, we only study the 2D version. It's valuable to extend this to a 3D version. Since the system is trained and tested with one specific camera, it's necessary to figure out a way to extend this system to other cameras with different parameters, or even to the same camera but with slight differences caused by manufacturing factors. In the domain of indoor localization, no single technology dominates, since different kinds of technologies have their own advantages over others. In addition, several measures can be taken to perfect the proposed system. Firstly, it's helpful to create more training data. Both distance and angle are continuous values. However, for the distance, only the data that locates on the limited locations are given, like 2m , 2.5m, 3m, etc. For the data of the angle, 15° is used to sample the data from 30° to 150°. The potential method to improve the performance is to increase the sampling locations for both distance and angles (e.g., the 5° could be used to sample the angle data). Secondly, as the angle is related to the shape instead of the largeness of the tag, it's possible to up-sample the tag when the tag is small or is taken from far away, to increase its resolution and, therefore, help the CNN perform better. Thirdly, it would be helpful to increase the measurement accuracy of the data, like the real distance or angle of the sampled data, as the errors in Figs. (4.14) and (4.15) are not consistent, especially for the error of the angle (e.g., the mean of absolute error for the angle from 40° to 80° jumps between 0° and around 4°, which is more likely caused by measurement errors).

Bibliography

[1] W. Bejuri, M. Mohamad and M. Sapri, “Ubiquitous Positioning: A taxonomy for location determination on mobile navigation system,” *Signal and Image Processing: An International Journal*, vol. 2, no. 1, pp. 24–34, Feb. 2011.

[2] P. Bahl and V. N. Padmanabhan, “RADAR: An in-building RF-based user location and tracking system,” in *Proc. IEEE INFOCOM 00*, vol. 2, Mar. 2000, pp. 775–784.

[3] R. Bruno and F. Delmastro, “Design and analysis of a Bluetooth-based indoor localization system,” Personal Wireless Communications, 2003, pp. 711725.

[4] T. Sattler, B. Leibe, and L. Kobbelt, “Fast image-based localization using direct 2D-to-3D matching,” in *Proc. IEEE ICCV’11*, Jul. 2011, pp. 667–674.

[5] J. Chung, M. Donahoe, C. Schmandt and I. Kim, “Indoor location sensing using Geo-Magnetism,” in *Proc. IEEE MobiSys11*, Jul. 2011, pp. 141–157.

[6] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, “Epsilon: A visible light based positioning system,” in *USENIX Conference NSDI11*, Jul. 2014, pp. 331–343.

[7] W. Bejuri, M. Mohamad, R. Zahilah, “A proposal of emergency rescue location (ERL) using optimization of inertial measurement unit (IMU) based pedestrian simultaneously localization and mapping (SLAM),” in *International Journal of Smart Home*, pp. 9-12, Jul. 2015.

[8] Z. Su, G. Shao, and H. Liu, “A soft-minimum method for NLOS error mitigation in TOA systems,” in *Proc. IEEE VTC’84*, Sept. 2016.

[9] Z. Su and H. Liu, “NLOS error mitigation with bias iterative deduction scheme in TOA based localization,” in *Proc. IEEE VTC’85*, Mar. 2017, (submitting).

[10] Y. He, A. Behnad, and X. Wang, "Accuracy analysis of the two-reference-node angle-of-arrival localization system," *IEEE Wireless Commun. Lett.*, vol. 4, no. 3, pp. 329–332, Jun. 2015.

[11] N. Patwari, I. Hero, A. O. M. Perkins, N. S. Correal and R. J. O. Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[12] N. Ravi, P. Shankar, A. Frankel, A. Elgammal and L. Iftode, "Indoor localization using camera phones," in *Proc. IEEE MCSA'06*, Apr. 2006.

[13] L. Chen, E. Wu, and G. Chen, "Intelligent fusion of Wi-Fi and inertial sensor-based positioning systems for indoor pedestrian navigation," *IEEE Sensors J.*, vol PP, issue 99, June 2014.

[14] S. Krishnan, P. Sharma, G. Zhang, and O. H. Woon, "A UWB based localization system for indoor robot navigation," in *Proc. IEEE ICUWB'07*, Sep. 2007, pp. 77-82.

[15] Q. Liu, Y. Wang, and A. E. Fathy, "Towards low cost, high speed data sampling module for multifunctional real-time UWB radar," *IEEE Trans. AES*, vol. 49, no. 2, Apr. 2013.

[16] R. C. Qiu, H. Liu, and X. Shen, "Ultra-wideband for multiple access communications," *IEEE Commun. Mag.*, vol. 43, no. 4, pp. 80-87, Feb. 2005.

[17] R. Ye, S. Redfield, and H. Liu, "High-precision indoor UWB localization: Technical challenges and method," in *Proc. IEEE ICUWB'10*, Sep. 2010.

[18] Baek, "Receiver synchronization for UWB TDOA localization," Master Thesis, Oregon State University, Aug. 2010.

[19] National Semiconductor: ADC083000, 8-Bit, 3 GSPS, High Performance, Low Power A/D Converter, Datasheet, 2008.

[20] Maxim Integrated Products: Synchronizing Multiple High-Speed Multiplexed DACs for Transmit Applications, Application Note 3901, 2006.

[21] L. Guoman, L. Yang, G. Meiguo, and S. Yiwei, “A method of synchronization between high-speed DAC chips,” in *Proc. IEEE ICISE’09*, 2009.

[22] T. Ohlemueller and M. Petri, “Sample synchronization of multiple multiplexed DA and AD converters in FPGAs,” in *Proc. IEEE Int. Symp. DDECS’11*, 2011.

[23] Xilinx FPGA Produces: Vertex-4 xc4vsx35 FPGA, Datasheet, 2008.

[24] ChipScope Pro Software and Cores (from Xilinx) : User Guide, 2012.

[25] Z. Wu, Y. Han, Y. Chen, and K.J.R. Liu, “A time-reversal paradigm for indoor positioning system,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1331–1339, Feb. 2015.

[26] C. Wu, Z. Yang, Y. Xu, Y. Zhao, and Y. Liu, “Human mobility enhances global positioning accuracy for mobile phone localization,” *IEEE Trans. Parallel Distrib. Syst*, vol. 26, no. 1, pp. 131–141, Jan. 2015.

[27] J. Hua, L. Meng, K. Zhou, B. Jiang, and D. Wang, “Accurate and simple wireless localizations based on time product of arrival in the DDM-NLOS propagation environment,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 2, pp. 239–246, Mar. 2015.

[28] C. Soares, J. Xavier, and J. Gomes, “Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements,” *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4532–4543, Sep. 2015.

[29] T. Nguyen, Y. Jeong, H. Shin, and M.Z. Win, “Machine learning for wideband localization,” *IEEE Trans. J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1357–1380, May 2015.

[30] I.T. Haque and C. Assi, “Profiling-based indoor localization schemes,” *IEEE Syst. J.*, vol. 9, no. 1, pp. 76–85, Sep. 2015.

[31] Z. Su, J. Huschle, S. Redfield, T. Qiao, F. Liu, and H. Liu, “High-speed real-time multi-channel data-acquisition unit: challenges and results,” in *Proc. IEEE CCNC’11*, Jan. 2014, pp. 105–112.

[32] J. Xu, M. Ma, and C. Law, “Position estimation using UWB TDOA measurements,” in *Proc. IEEE ICUWB*, Sep. 2006, pp. 605–610.

[33] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: A survey, *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, Jul. 2004.

[34] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” in *Proc. OSDI’02*, vol. 36, 2002, pp. 147–163.

[35] Y. Yang and K. Yang, “Time synchronization for wireless sensor networks using the principle of radar systems and UWB signals,” in *Proc. IEEE Information Acquisition*, Aug. 2006, pp. 160–165.

[36] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, “TDOA location system for IEEE 802.11b WLAN,” in *Proc. IEEE WCNC*, vol. 4, Mar. 2005, pp. 2338–2343.

[37] J. Liu, Z. Wang, and J. Cui, “A joint time synchronization and localization design for mobile underwater sensor networks,” *IEEE Trans. Mobile Comput.*, vol. PP, no. 99, pp. 1–14, April 2015.

[38] O. Jean and A.J. Weiss, “Passive localization and synchronization using arbitrary signals,” *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 2143–2150, Feb. 2014.

[39] R.T. Rajan and A.J. Van Der Veen, “Joint ranging and synchronization for an anchorless network of mobile nodes,” *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 1925–1940, Mar. 2015.

[40] S. Ganeriwal, R. Kumar, and M.B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proc. SENSYS’1*, Mar. 2003, pp. 138–149.

[41] P. Sommer and R. Wattenhofer, “Gradient clock synchronization in wireless sensor networks,” in *Proc. IEEE IPSN’09*, Mar. 2004, pp. 37–48.

[42] B. Xu, G. Sun, R. Yu, and Z. Yang, “High-accuracy TDOA-based localization without time synchronization,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1567–1576, Aug. 2013.

[43] S.M. Kay, “Fundamentals of statistical signal processing, volume I: Estimation theory,” *Prentice Hall*, 1993.

[44] S. Venkatesh and R.M. Buehrer, “NLOS mitigation using linear programming in ultrawideband location-aware networks,” *IEEE Trans. Vehicular Technology*, vol. 56, no. 5, pp. 3182–3198, Sep. 2003.

[45] Y.T. Chan and K.C. Ho, “A simple and efficient estimator for hyperbolic location,” *IEEE Trans. Signal Process*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.

[46] R.M. Vaghefi and M.R. Gholami, “Bearing-only target localization with uncertainties in observer position,” in *Proc. IEEE PIMRC’21*, Sep. 2010, pp. 238–242.

[47] J. Niu, B. Wang, L. Shu, T.Q. Duong, and Y. Chen “ZIL: An energy-efficient indoor localization system using ZigBee radio to detect WiFi fingerprints,” *IEEE Trans. Selected Areas in Communications*, vol. 33, no. 7, pp. 1431–1442, May. 2015.

[48] I. Guvenc, C. Chong, and F. Watanabe, “NLOS identification and weighted least squares localization for UWB systems using multipath channel statistics.” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 7, pp. 1–14, Aug. 2007.

[49] K. W. K. Lui, H. C. So, and W.-K. Ma, “Maximum a posteriori approach to time-of-arrival-based localization in non-line-of-sight environment,” IEEE Trans. Veh. Technol., vol. 59, no. 3, Mar. 2010.

[50] G. Wang, Y. Li, and R. Wang, “A new semidefinite relaxation method for acoustic energy based source localization, IEEE Sensors J., vol. 13, no. 5, May 2013.

[51] E.R. Jativa, D. Sanchez, and J. Vidal, “NLOS mitigation based on TOA for mobile subscriber positioning systems by weighting measures and geometrical restrictions,” in *Proc. IEEE APCASE*, Jul. 2015, pp. 325–330.

[52] S. Zhang, S. Gao, G. Wang, and Y. Li, “Robust NLOS error mitigation method for TOA-based localization via second-order cone relaxation,” *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–4, Sep. 2015.

[53] M. Horiba, E. Okamoto, T. Shinohara, and K. Matsumura, “An improved NLOS detection scheme using stochastic characteristics for indoor localization,” in *Proc. IEEE ICOIN*, Jan. 2015, pp. 478–482.

[54] R. Vaghefi, J. Schloemann, and R. Buehrer, “NLOS mitigation in TOA-based localization using semidefinite programming,” in *Proc. IEEE WPNC’10*, Mar. 2013, pp. 1–6.

[55] G. Wang, H. Chen, Y. Li and N. Ansari, “NLOS error mitigation for TOA-based localization via convex relaxation,” *IEEE Trans. Wireless Communications*, vol. 13, no. 8, pp. 4119–4131, Aug. 2013.

[56] R.M. Vaghefi and R.M. Buehrer, “Cooperative localization in NLOS environments using semidefinite programming,” *IEEE Communications Letters*, vol. 19, no. 8, pp. 1382–1385, Aug. 2015.

[57] T. Nguyen, J. Youngmin and H. Shin and M.Z. Win, “Machine learning for wideband localization,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1357–1380, May. 2015.

[58] S. Venkatesh and R. M. Buehrer, "NLOS mitigation using linear programming in ultra wideband location-aware networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3182–3198, Sep. 2007.

[59] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.

[60] R. M. Vaghefi and M. R. Gholami, "Bearing-only target localization with uncertainties in observer position," in *Proc. IEEE PIMRC*, Sep. 2010, pp. 238–242.

[61] B. Li, A. Dempster, C. Rizos and H. K. Lee, "A database method to mitigate NLOS error in mobile phone positioning," in *Proc. IEEE PLANS*, Apr. 2006, pp. 173–178.

[62] J. Schroeder, S. Galler and K. Kyamakya, "NLOS detection algorithms for ultra-wideband localization." in *Proc. IEEE WPNC'4*, Apr. 2007, pp. 159–166.

[63] J. Borras, P. Hatrack and N. Mandayam, "Decision theoretic framework for NLOS identification," in *Proc. IEEE VTC'48*, vol. 2, May 1998, pp. 1583–158.

[64] I. Guvenc, C. C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *Proc. IEEE WCNC'07*, Mar. 2007, pp. 1571–1576.

[65] X. Wang, Z. Wang, and B. O. Dea, "A TOA based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 112–116, Jan. 2003.

[66] K. Yu and Y. G. Guo, "Improved positioning algorithms for nonline-of-sight environments," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2342–2353, Jul. 2008.

[67] K. W. K. Lui, H. C. So, and W. K. Ma, "Maximum a posteriori approach to time-of-arrival-based localization in non-line-of-sight environment," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1517–1523, Mar. 2010.

[68] E. R. Jativa, D. Sanchez, and J. Vidal, "NLOS mitigation based on TOA for mobile subscriber positioning systems by weighting measures and geometrical restrictions," in *Proc. IEEE APCASE*, Jul. 2015, pp. 325–330.

[69] Y. Yu and L. H. Zhang, "WSN location method based on BP neural network in NLOS environment," in *Proc. IEEE WCSN*, Dec. 2014, pp. 321–325.

[70] T. Nguyen, J. Youngmin, and H. Shin and M. Z. Win, "Machine learning for wideband localization," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1357–1380, May. 2015.

[71] G. Wang, H. Chen, Y. Li, and N. Ansari, "NLOS error mitigation for TOA-based localization via convex relaxation, *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4119–4131, Aug. 2014.

[72] S. Zhang, S. Gao, G. Wang, and Y. Li, "Robust NLOS error mitigation method for TOA-based localization via second-order cone relaxation," *IEEE Commun. Lett.*, vol. 19, no. 12, pp. 2210–2213, Dec. 2015.

[73] R. Vaghefi, J. Schloemann, and R. Buehrer, "NLOS mitigation in TOA-based localization using semidefinite programming," in *Proc. IEEE WCNC'10*, Mar. 2013, pp. 1–6.

[74] R. Vaghefi and R. Buehrer, "Cooperative localization in NLOS environments using semidefinite programming," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1382–1385, Aug. 2015.

[75] W. Richard, G. Uwe, M. Oliver, and R. Julia, "WSN-based passenger localization in severe NLOS environments using SDP," in *Proc. IEEE IPIN*, Oct. 2013, pp. 1–7.

[76] S. Venkatesh and R. M. Buehrer, "NLOS mitigation using linear programming in ultra-wideband location-aware networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3182–3198, Sep. 2007.

[77] T, Qiao, S. Redfield, A. Abbasi, Z. Su, and H. Liu, "Robust coarse position estimation for TDOA localization," *IEEE Wireless Commun. Lett.*,vol. 2, no. 12, pp. 623–626, Sept. 2013.

[78] J. Schroeder, S. Galler, and K. Kyamakya, "NLOS detection algorithms for ultra-wideband localization," in *Proc. IEEE WPNC'04*, Apr. 2007, pp. 159–166.

[79] R. W. Ouyang and A. K. S. Wong, "An enhanced TOA-based wireless location estimation algorithm for dense NLOS environments," in *Proc. IEEE WCNC*, Apr. 2009, pp. 1–6.

[80] I. Guvenc, C. C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *Proc. IEEE WCNC*, Mar. 2007, pp. 1571–1576.

[81] X. Wang, Z. Wang, and B. O. Dea, "A TOA based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 112–116, Jan. 2003.

[82] K. Yu and Y. G. Guo, "Improved positioning algorithms for nonline-of-sight environments," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2342–2353, Jul. 2008.

[83] T. Qiao and H. Liu, "Improved least median of squares localization for Non-Line-of-Sight mitigation," *IEEE Wireless Commun. Lett.*, vol. 2, no. 12, pp. 1451–1454, Sept. 2014.

[84] Z. Cai, X. Yang, and X. Wu, "Tetrahedron-constraint least square localization algorithm in mixed LOS/NLOS scenario based on TDOA measurements," in *Proc. IEEE IS3C'16*, Aug. 2016.

[85] A. Gaber and A. Omar, “Utilization of multiple-antenna multicarrier systems and NLOS mitigation for accurate wireless indoor positioning,” *IEEE Trans. Wireless comm.*, vol. 15, no. 10, pp. 6570–6584, Jul. 2016.

[86] L. Zhang, H. Ma, and C. Sun, “A novel indoor intelligent location algorithm based on GA-BFO,” in *Proc. IEEE ICISCE’03*, Jul. 2016.

[87] Z. Abu-Shaban, X. Zhou, and T. Abhayapala, “A novel TOA-based mobile localization technique under mixed LOS/NLOS conditions for cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8841–8853, Mar. 2016.

[88] S. Zhang, S. Gao, G. Wang, and Y. Li, “Robust NLOS error mitigation method for TOA-based localization via second-order cone relaxation,” *IEEE Commun. Lett.*, vol. 19, no. 12, pp. 2210–2213, Dec. 2015.

[89] R. Vaghefi, J. Schloemann, and R. Buehrer, “NLOS mitigation in TOA-based localization using semidefinite programming,” in *Proc. IEEE WCNC’10*, Mar. 2013, pp. 1–6.

[90] R. Vaghefi and R. Buehrer, “Cooperative localization in NLOS environments using semidefinite programming,” *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1382–1385, Aug. 2015.

[91] G. Wang, A. Man-Cho So, and Y. Li, “Robust convex approximation methods for TDOA-based localization under NLOS conditions,” *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3281–3295, Jul. 2016.

[92] K. M. Al-Qahtani, A. S. Al-Ahmari, A. H. Muqaibel, M. A. Landolsi, and U. M. Johar ,“Improved residual weighting for NLOS mitigation in TDOA-based UWB positioning systems,” in *Proc. IEEE ICT*, May. 2014, pp. 211–215.

[93] K. Lee, J. Oh, and K. You, “TDOA/AOA based geolocation using Newton method under NLOS environment,” in *Proc. IEEE ICCCBDA*, Jul. 2016.

[94] K. Yang, G. Wang, and Z. Luo, "Efficient convex relaxation methods for robust target localization by a sensor network using time difference of arrivals," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2775–2784, Jul. 2009.

[95] B. Y. Shikur and T. Weber, "TDOA/AOD/AOA localization in NLOS environments," in *Proc. IEEE ICASSP*, May 2014, pp. 6518–6522.

[96] S. Li, M. Hedley, I. B. Collings, and D. Humphrey, "TDOA-based localization for semi-static targets in NLOS environments," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 2162–2337, Jun. 2015.

[97] X. Liu, H. Makino, and K. Mase, "Improved indoor location estimation using fluorescent light communication system with a nine-channel receiver," *IEICE Trans. Commun.*, vol. 93, no. 11, pp. 2936–2944, Aug. 2010.

[98] Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation," in *Proc. IEEE ICC'02*, Apr. 2002, pp. 436–439.

[99] J. Biswas and M. Veloso, "WiFi localization and navigation for autonomous indoor mobile robots," in *Proc. IEEE ICC'10*, Apr. 2010, pp. 4379–4384.

[100] H. Iglesias, V. Barral, and C. Escudero, "Indoor person localization system through RSSI Bluetooth fingerprinting," in *Proc. IEEE IWSSIP'12*, Apr. 2012, pp. 40–43.

[101] S. Chawathe, "Low-latency indoor localization using Bluetooth beacons," in *Proc. IEEE ITS'12*, May 2012, pp. 334–340.

[102] R. Sriram and D. Jalihal, "TDoA based EKF localization for LTE." in *Proc. IEEE NCC'16*, May 2016, pp. 1–6.

[103] J. Chung, M. Donahoe, C. Schmandt, I. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism." in *Proc. IEEE ICMSAS'9*, Apr. 2011, pp. 141–154.

[104] Y. Zhao, K. Liu, Y. Ma, Z. Gao, Y. Zang, and J.Teng, “Similarity analysis-based indoor localization algorithm with backscatter information of passive UHF RFID tags,” *IEEE J. Sensors*, vol. 17, no. 1, pp. 185–193, Jan. 2017.

[105] R. Li, Z. Du, Y. Zhao, and S. Liu, “Design and implementation of mobile robot ultrasonic localization system,” in *Proc. IEEE CCDC’16*, Jul. 2016, pp. 5347–5352.

[106] R. Mautz, “Indoor positioning technologies,” Doctoral dissertation, *Institute of Geodesy and Photogrammetry*, 2012.

[107] D. Sinha, M. Ahmed, and M. Greenspan, “Image retrieval using landmark indexing for indoor navigation,” in *Proc. IEEE CCCRV’14*, Jun. 2014, pp. 63–70.

[108] Y. Huang, H. Wang, K. Zhan, J. Zhao, P. Gui and T. Feng, “Image-based localization for indoor environment using mobile phone,” *ISPRS*, vol. 4, pp. 211–215, May 2015.

[109] H. Bay and T. Tuytelaars, “Speeded up robust features,” *ECCV Computer Vision*, vol. 4, pp. 404–417, May 2006.

[110] J. Huang, S. Lee, and C. Tsai, “A fast image matching technique for the panoramic-based localization,” in *Proc. IEEE ACIS’16*, Jun. 2016, pp. 1–6.

[111] Q. Anwar, A. Malik, and B. Thrnberg, “Design of coded reference labels for indoor optical navigation using monocular camera,” in *Proc. IEEE ICIPIN’13*, Oct. 2013, pp. 112–120.

[112] V. Badrinarayanan, A. Kendall, and R. Cipolla, “A deep convolutional encoder-decoder architecture for image segmentation,” arXiv preprint arXiv:1511.00561, 2016.

[113] Template Matching, OpenCV: ”http://docs.opencv.org/2.4/doc/tutorials /imgproc/histograms/template_matching/template_matching.html”

[114] D. Xu and Y.Tian, “A comprehensive survey of clustering algorithms,” *Springer Annals of Data Science*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[115] F. Li, C. Zhao, G. Ding, and F. Zhao, “A reliable and accurate indoor localization method using phone inertial sensors,” in *Proc. ACM UbiComp’12*, Sept. 2012, pp. 421–430.

[116] G, Kim and E. M. Petriu, “Fiducial marker indoor localization with Artificial Neural Network,” in *Proc. IEEE AIM’10*, Oct. 2010.

[117] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. “SURF: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 3, no. 2, pp. 346–359, Jun. 2008.

[118] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Intl. J. of Computer Vision*, vol. 2, pp. 91–110, 2004.

[119] Sergi Arias, “Visual tag recognition for indoor positioning,” Masters Thesis at KTH, 2013.

[120] D. Tax, “One class classification,” PhD thesis, Delft University of Technology 2001.