Toward an Accurate Acoustic Localization System

Mohamadbagher Fotouhi

A Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMP SCIENCE AND SYSTEMS

University of Washington

2019

Program Authorized to Offer Degree:
Computer Science and System

University of Washington

Abstract

Toward
an Accurate Acoustic
Localization System

Mohamadbagher Fotouhi

Chair of the Supervisory Committee:
Assistant Professor Wei Cheng

In this dissertation, we propose an accurate and fast multi-pair simultaneous localization systems for smartphones without the need of infrastructure support. The system is a purely software-based solution (an App), which uses only the basic set of commodity smartphone capabilities including speakers, microphones, and Wi-Fi/Bluetooth. Audible-band acoustic signals and Wi-Fi/Bluetooth are jointly utilized for time of arrival (TOA)-based ranging. To simultaneously offer multi-smartphone-pair ranging, the acoustic chirps are encoded by Gold sequences, which has been extensively used in Code Division Multiple Access (CDMA) systems. We designed and implemented a fast One-way acoustic communication multi-pair ranging system as well as more accurate two-way acoustic communication multi-pair ranging system for comparison. Obtained ranges then will be used as input to a range-based localization approach in which the mobile devices can localize themselves related to each other. Our design is scalable with respect to the number of devices. The system is server-less and mobile devices compute their position locally so our system improve the user privacy. Through extensive experimental evaluations over 8 different smartphones, the proposed system's positioning errors are around one meter and ranging errors are only a few centimeters for the actual phone-to-phone distances up to 60 meters in both indoor and outdoor environments.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

DEDICATION

To Dad, Who took me to the library.

I miss you.

Chapter 1

INTRODUCTION

There is no doubt that in recent days mobile devices and smartphones have turned into the most popular individualized computing devices. A large variety of applications require the user's actual location to personalize and customize their services. In GPS-denied environments, a user's location is obtained by utilizing distance information. Typical applications, which need device-to-device distance information, such as in-door way-finding, air drone swarm control, and autonomous underwater vehicle (AUV) localization. Many approaches have been developed to address the device-to-device distance ranging problem. There are Radio Frequency-based (Rf) solutions, acoustic-based ones, and the ones leveraging specific hardware such as ultrasound, self-calibrating acoustic platform, Pinpoint system, and channel state indicator (CSI). In general, acoustic based solutions are more accurate than RF-based ones for off-the-shelf smartphones, where other specific hardware is not available. Between the existing acoustic smartphone distance ranging/positioning Beep-Beep[25, 8] could achieve the best accuracy, however it is sometimes impractical:

- Its delay may not be good for real-time applications (more than 5 second delay).
- It has a high system overhead.
- It is not precise if the target is moving, which is generally the case in mobile networks.
- It is not able to support multiple devices or multiple pairs simultaneous ranging.
- It may need centralized computations [8].
- It has limited range (5 meter).
- It just provide the distance between two devices and doesn't perform localization.

If several smartphones or drones try to use Beep-Beep, to prevent acoustic signal collision each pair should run this algorithm separately. As Beep-Beep usually takes more than 5 seconds for each pair ranging, in case of the users are moving, it is impractical because this delay will cause a big error.

In this paper we are presenting a fast real-time acoustic localization system which can support multiple devices and multiple pairs simultaneously without the need of centralized computations. Our solution is purely software-based (an APP) installed without root permission for large-scale deployment. It does not need any extra hardware and utilizes only the most basic built in devices (microphones, speakers, and Wi-Fi/Bluetooth), which are already incorporated in all of the commercial devices. Our system can be easily extended to other application scenarios such air drone swarm. Technically, we jointly use Wi-Fi/Bluetooth, audio communications, and code-division multiple access (CDMA) to estimate the distances between the phones and then using the distance information each mobile device can localize itself locally relative to other devices. Our technical contributions and innovations are summarized as follows:

- We propose Single-chirp ranging as a one way acoustic ranging method to provide real-time ranging services with 10-50 cm accuracy in 0.5 seconds.

- We have identified and designed the most efficient shape for the acoustic signal and detection scheme which have not been used for acoustic ranging to the best of our knowledge. Our acoustic signal is enabling it to support multiple smartphone-pair ranging simultaneously.

- We proposed Double-chirp ranging as a two way acoustic ranging method which has centimeter accuracy. Comparing to other two way acoustic ranging methods ours method is faster and more accurate, it can provide centimeter accuracy in less than 2 seconds for each pair while it can support multiple device/multiple pairs simultaneously.

- We have implemented a time efficient range base localization approach [5] to calculate relative position of each device using the ranging information.

- We have implemented our solution on Android smartphones and evaluated the performance in both indoor and outdoor environments using different models of smartphones.

This dissertation is organized as follows: In Sec. 2, the most related works are summarized. In Sec. 3 the challenges in phone-to-phone ranging and the key ideas of our corresponding solutions are discussed. The detailed signal design and detection are introduced in Sec. 4. In Sec. 5, the communication protocol design for phone-to-phone ranging is described. A Single-chirp ranging approach is studied in Sec. 6 along with Double-chirp approach. In Sec.6.3 we have explained our localization approach. The evaluation results are reported in Sec. 7. Finally, in Sec. 8, this paper is concluded and our future work is discussed.

Chapter 2

RELATED WORK

Many range-base localization systems have been designed for indoor localization [1, 14, 21, 8]. They can be categorized by the techniques (RF and acoustic) used for calculating the distance. RF-based methods are generally less accurate such as PinPoint [35] (4-6 feet). We focus on the acoustic-based ones in this paper.

Most of the highly accurate ranging systems rely on ultrasound [24]. TOA systems compute distances based on the single flying time between a pair of sender and receiver. The Bat system [9, 2] is an example of these systems. In Bat, ultrasonic tags are used to identify users and objects, which were called “bats”. These bats emit periodic ultrasonic signals to the receivers mounted across the ceiling. The reported accuracy is almost 3 cm. Time-Difference-of-Arrival (TDOA) systems can remove the requirement of knowing exactly when a signal was transmitted via using what is known as pseudo-ranging [15]. The Dolphin is an ultrasonic positioning system that adopts the pseudo-ranging approach [10, 11]. It attaches sensors to various indoor objects, to send and receive RF and ultrasonic signals. The reported accuracy is 2 cm in the range of almost 10 m. Cricket [27] uses communication mediums with two propagation velocities, specifically RF radio and ultrasonic signals, to achieve 2 cm accuracy in 10 m range. The Cricket nodes are small ultrasonic devices. Whistle [33] is another TDOA based system, which is a synchronization-free localization system. The reported results show 10-20 cm accuracy in almost 10 m range. Whistle leverages several stable receivers, whose locations are known, to receive two sound signals from a target for ranging and localization. In WALRUS [3], ultrasound signal emitted from PDAs or Laptops at a frequency of 21 kHz is used to identify their location within a room. It can achieve a room-level precision of mobile device location with ultrasound. The limitations of applying these ultrasonic techniques are from the requirements of deploying a large number of receivers and ultrasonic enabled devices, which do not include today’s smartphones. In

[8], each smartphone uses Double-chirp and sends all the collected acoustic signals to a central server for distance estimation. The server will then notify each smartphone about their locations.

There are a few approaches, which need special devices for non-ultrasonic band acoustic ranging. Guoguo [17] uses a special hardware to transmit the beacons and the doublet pulse between 15 and 20 kHz. The Hadamard codes were used to identify the location of a smartphone. Beep [18] is TOA-based localization system, which finds the location of a sound source in a 3D room with the accuracy of 60 cm. ASSIST [13] uses special devices to receive the acoustic chirps (18kHz - 21kHz) transmitted from smartphones for ranging. In [4], special devices with ZigBee modules use TDOA to find their distances from a sound source.

For off-the-shelf smartphones, Beep-Beep [26, 29], which uses Round-Trip-Time-of-Flight (RTOF), is a common method utilized in software ranging. Beep-Beep [26] is the most related work of ours. It uses a chirp pulse between 2 kHz and 6 kHz in two-way acoustic communication. A matched filter is used to calculate the Estimated-Time-Of-Arrival (ETOA). It can achieve the accuracy of 3-5 cm in range of 5 m. Beep-Beep, however, does not scale well beyond 1 pair of devices as the same chirp used across multiple devices may result in interference during ranging. Moreover, it is not suitable for mobile environments as its processing delay is high it's effective range is short and it does not provide any position information.

Compared to existing works, we target on a unique challenge of fast and accurate localization among multiple pairs of smartphones. We have designed and identified an efficient shape of the acoustic signal, which is robust to noisy environments and multi-path effect. CDMA is used to encode our acoustic signal to avoid signal interference and collision. Our signal design, signal detection, and ranging approaches are lightweight in terms of computation and communication. In a word, we enable multi-pair-smartphone ranging with centimeter-level accuracy in less time and at longer range. We have identified a time efficient server-less localization framework to implement ranging data and perform a fast accurate positioning.

Chapter 3

CHALLENGES AND SOLUTION

Leveraging acoustic signals and using time of arrival (TOA) technique is a promising method to achieve high ranging accuracy, but it has its own challenges. we will describe the main factors which we focus on and our approaches for solving them in this section.

Using acoustic signals and working in indoor environments, noisy, fading, and multi-path channels will be one of the main problems. Our solution uses a chirp signal which can overcome or at least mitigate this problem. In addition, we want our system to support multiple users. The main concern is to differentiate the users and mitigate the interference among their acoustic signals. Our solution for this challenge uses the CDMA Gold code to encode the chirp signal. Finally, the Time-of-Arrival estimates will be accurate if both devices' clocks are synchronized. We use Wi-Fi/Bluetooth signals for clock synchronization in Single-chirp and we implemented the Double-chirp to avoid clock synchronization. We consider our system as a time-critical positioning service. Providing localization using conventional triangulation methods are unfortunately unsuitable for our system, for these methods neglect the time needed for localization. To address this issue we have used ODFL (On Demand Fast Localization) method introduced in [5].

3.1 Using Chirp Signals

During its propagation, the acoustic signal will lose part of its energy. As a result, the received signal will be attenuated at the receiver after traveling through the air. In our design, we consider the environment to be polluted by colored ambient noise. Therefore, while propagating to the receiver, not only the signal is losing its energy but also there exists some noise in every frequency range, which can pollute the signal and makes it difficult to detect.

In addition to signal attenuation and noise effect in indoor environments, we have to deal with the reverberation, the effect of the reflected signals from walls, roof, floor, or any object that might be around the transmitter/receiver or in between causes multi-path problem. A suitable acoustic signal for our system should have three properties: 1) We should be able to play and record it using the smart-devices commodity. 2) We should be able to distinguish the signal from background noise. 3) The signal should not be similar to a time-shifted version of itself. To address this challenge, we use a chirp signal instead of a simple beep. A chirp is a linearly frequency modulated sinusoidal signal, whose frequency changes over time. The chirp signals have very good auto-correlation characteristics, and they are widely used in radar applications [6], localization [16] , and acoustic ranging [15]. Since the frequency of a chirp signal is changing linearly over time, the multi-path chirp-signal components will appear without significant mutual interference at the matched-filter output [12]. Thus the chirp signal provides an excellent protection against signal fading due to the multi-path propagation.

Table 3.1: Chirp Properties

$Starting\ frequency$	f_1	$2kHz$
$Ending\ frequncey$	f_2	$6kHz$
$Duratoin$	t_e	$50ms$
$SampleRate$		$48kHz$

We design an acoustic chirp to occupy a specific range of frequencies that matches the device's speaker and microphone capabilities. Moreover, the duration of the chirp has been chosen to be small enough to make the multi-path effect at the receiver side negligible. The properties of the chirp are shown in Table 3.1.

3.2 Using CDMA for Encoding the Chirp

Our goal is to design a system which can support ranging among multiple pairs of smartphones. In other words, several users can estimate their distances from one to another simultaneously. To support multiple users, we need to be able to differentiate the audio signals sent by different devices. We use CDMA Gold code to encrypt audio signals. Since this CDMA code needs to be known by both devices, it is sent through Wi-Fi just before audio signal is sent. After receiving the audio signal, a matched filter is adapted to decrypt the signal via using the CDMA code. Then, the time of arrival (TOA) of the signal is estimated.

3.3 Time efficient localization

The fundamental flaw in conventional triangulation localization approach is that it requires the information source and destination to be localized under the same coordinate system before any information transmission can be initiated. After all conventional approach relies on flooding in the network to disseminate location information. This will make all the devices in the network to participate in the location updating and forwarding process which will cause a heavy overhead. In order to address this problem we decided to use the ODFL approach. In ODFL localization starts from multi-coordinate systems. ODFL captures the earliest time for starting localization or location base service (LBS) applications. Each intermediate user on the router can help transforming the information of the position of the source to the one in its next-hop user's coordinate system until the destination successfully obtains the position of the source in its own coordinate system.

3.4 Clock Synchronization

In Single-chirp one device sends a Wi-Fi message containing its time to set the clock of the other device in reference to the first one. In this way, the clocks of two devices are synchronized. In this approach, the devices will synchronize their clocks just once at the beginning of the ranging process. Then, the ranging can be done just by sending one audio signal (Single-chirp).We also developed and implemented the Double-chirp approach, which is a two-way communication model to avoid clock synchronization between two devices. We study both of the approaches and analyze their results in terms of ranging time-complexity, reliability, and accuracy.

3.5 Ranging System Model

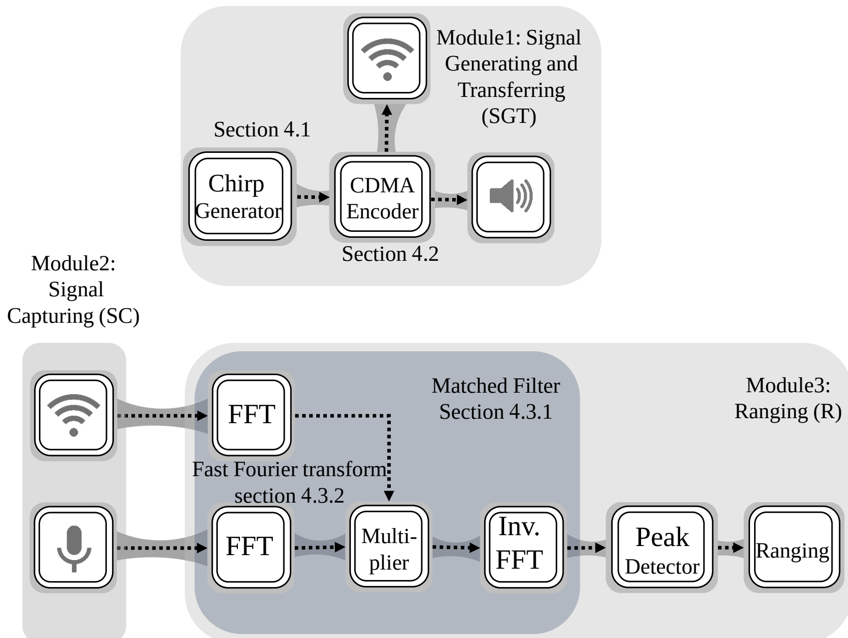


Figure 3.1: System Model

Our Ranging system model is illustrated in Fig. 3.1. We have defined three modules in our system, Signal Generating and Transferring (SGT) module, Signal Capturing (SC) module, and Ranging (R) module. Note that a device can have more than one modules. Since we need to run multiple modules simultaneously, multi-threading is used as shown in Table 3.2.

A device equipped with the SGT module generates an acoustic chirp and encodes it using Gold code. Then, it runs Thread 1 and Thread 3 to send the seed of the Gold code through Wi-Fi/Bluetooth, and plays the encoded acoustic signal with its speaker.

The device, which has the SC module, runs Thread 1 to receive the seed of the Gold code via Wi-Fi/BLT and reconstructs the encoded chirp as a reference signal. Then, it runs Thread 2 to record the received acoustic signal using its microphone.

A device with the Ranging module runs Thread 4 to use the matched filter to find the

Table 3.2: Threads

Thread No.	task
Thread 1	send/receive Gold code seeds via Wi-Fi/BLT
Thread 2	record audio signals
Thread 3	play audio signals
Thread 4	matched filtering

start time index of the signal (TOA), based on which it estimates the range from the SGT device. Note that we use fast Fourier transform (FFT) to improve the matched filtering processing efficiency.

Chapter 4
SIGNAL DESIGN AND DETECTION

In this section we present more in-depth details on how we use chirp signals to mitigate fading and multi-path effects. The chirp signal's auto-correlation property makes it easier to detect by using a matched filter (We demonstrate how we use matched filtering for accurate signal detection).

4.1 Acoustic Chirps

A system for estimating the TOA of a signal needs a reliable and accurate signal detector. If the received signal is corrupted by additive random noise, the matched filter [31] is the optimal linear filter, in the sense of maximizing the signal-to-noise ratio (SNR) at the receiver. It has been widely used in radar and sonar systems. Therefore, we choose the matched filter as the signal detector at the receiver. In the matched filter, the received acoustic signal is detected by its correlation with a reference acoustic signal, which has been reconstructed by the receiver using the Gold code sent from the transmitter through Wi-Fi/BLT. To have an accurate TOA estimate, it is critical that the signal possesses a very "sharp" autocorrelation function, i.e. the autocorrelation is very high when the time difference is zero and quickly attenuates when the time difference increases. In addition, as we discussed in Sec. 3.1, the signal should be robust to the multi-path and fading effects.

A frequency varied sinusoid signal (chirp) is one choice that meets all of our requirements. In our experiments, we use a finite duration linear chirp. The signal duration should not be too small to be detected, and it should not be too large to cause unwanted delay. We set the chirp signal duration to be 0.05 seconds. The frequency of the chirp signal should be limited in a range such that when the signal is passing through the microphone and speaker's filters, it does not get distorted. Based on the dynamic frequency range of most of the mobile devices' microphones, we design the chirp signal as follows:

The starting frequency and the ending frequency are set as 2 kHz and 6 kHz, denoted as f_1 and f_2, respectively. the sampling rate is 48 kHz, denoted as S_R. The signal duration is 0.05 seconds, denoted as t_e. Then, the slope of the signal, denoted as α, can be calculated by (4.1), which is the chirp rate.

$$\alpha = \frac{f_2 - f_1}{t_e} \tag{4.1}$$

For any t between 0 and t_e, the signal can be calculated as follows.

$$s(t) = \cos\left[2\pi\left(\frac{\alpha}{2}t^2 + f_1 t\right)\right] \tag{4.2}$$

The chirp signal generated by this function is shown in Fig. 4.1.a.

When we play the sound through the speaker, due to the speaker diaphragm inertia, the signal will get distorted in the first few milliseconds. To solve this issue, we add a 5-millisecond simple cosine signal with the frequency of 2 kHz. As a result, during the first few milliseconds, this cosine signal will be played. Then, after the speaker gets warmed up, the chirp signal will be played correctly. The waveform of the signal played is shown in Fig 4.1.b.

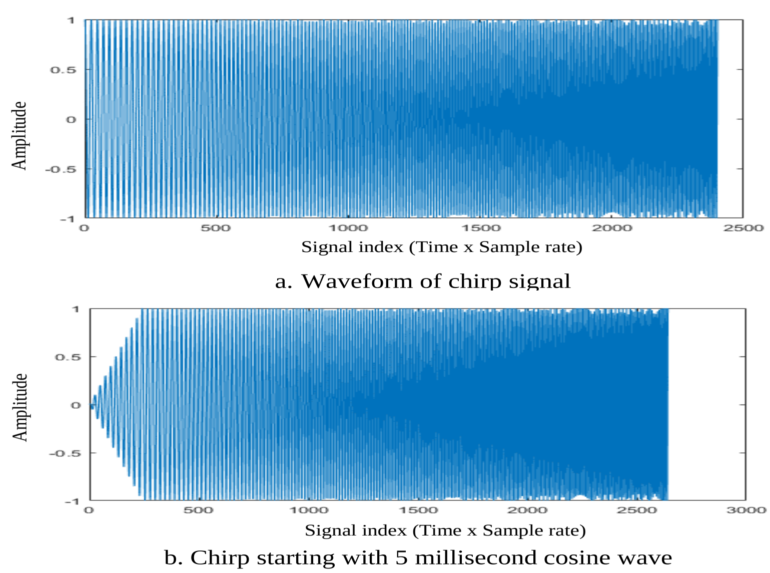


Figure 4.1: Waveform of the played signal

4.2 Supporting Multiple Smartphone Pairs

In our targeted application scenarios, multiple devices may simultaneously perform their ranging tasks. Thus, we face two challenges:

- When several devices try to find their distances from a single device, their acoustic signals should be distinguishable.

- When there are different pairs of devices that try to estimate their pair-wise ranges simultaneously, it is imperative that the interference between the devices' acoustic signals is minimum.

To prevent interference among the generated signals when multiple devices are broadcasting in the same frequency range, and to be able to distinguish the acoustic signals of different devices, the correlation between the signals has to be as small as possible while each of the acoustic signals should have good autocorrelation property to achieve accurate signal detection in the presence of ambient noise. One typical signal design that fits our requirement is Gold code signal as the Gold codes have small cross-correlation within their sets [7]. Gold codes have bounded small cross-correlations within a set, which is useful when multiple devices are broadcasting in the same frequency range. A set of Gold code sequences consists of $2^n - 1$ sequences each one with a period of $2n - 1$. In our system we use the degree $n = 6$, and the length of Gold sequence $N = 2^n + 1 = 63$. Thus, the $mSequence1 = z^5 + z^2 + 1$ and $mSequence2 = z^6 + z^5 + z^2 + z + 1$ [28] [20], where z is the initial sequence to separate the signal from other ones. Since degree $n = 6$, the seed value has to be less than $2^{12} - 1 = 4095$, so that the first initial sequence is the higher 6 bits of the binary value of seed and the second sequence is the lower 6 bits of the seed. We chose the seed to be 1020 and the number of the samples of the Gold sequence is 567. The two sequences are generated by the linear feedback shift operation using the above equation $z^5 + z^2 + 1$ and $z^6 + z^5 + z^2 + z + 1$. The Gold sequence is the XOR of mSequence1 and mSequence2. Using this CDMA signal about 3600 devices can communicate with minimum interference. Note that 3600 is much greater than the number of devices within a 4-hop phone-to-phone network for now.

4.3 Detection

The Ranging module (R) determines exactly when the acoustic signal arrives. Hence, it is critical to the accuracy of ranging. In this subsection, we describe the detection protocol used in our system.

4.3.1 Matched Filter

The detection of the received signal is done by a process called matched filter. Matched filtering is a pure software technique. The matched filter calculates the correlation between two signals, by convolving the recorded signal with a conjugated and time-reversed version of the reference signal that has been reconstructed using the CDMA Gold code. This correlation shows the similarity between two signals while they slide across each other. The highest cross-correlation of the signals indicates the most similarity. It is shown as a peak in the final result. By finding the peak, we will be able to determine the starting time of the signal. To use matched filter, the impulse response of the filter is generated with the mirrored signal of CDMA encoded chirp signal sequence. The output of the matched filter is given by the convolution as

$$y(t) = \int x(\lambda)h(t-\lambda)d\lambda \tag{4.3}$$

where $x(t)$ and $h(t)$ are the received signal and the impulse response of the matched filter respectively. In Fig. 4.2, the recorded signal and the output from the matched filter are shown.

4.3.2 Fast Signal Detection using FFT Convolution

The computational overhead of (4.3) is high. To support multiple users, the device with a Ranging module must calculate the time index from its recorded signal as fast as possible. It will be significantly more efficient if we use fast Fourier transform (FFT) to perform matched filtering in the frequency domain. FFT convolution uses the overlap-add method [22] in conjunction with the FFT, allowing signals to be convolved by multiplying their frequency

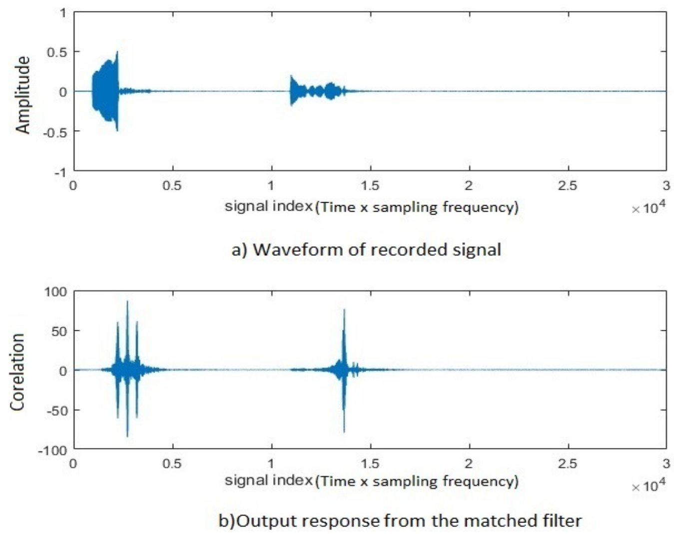


Figure 4.2: Recorded signal and matched filter output

spectra. The overlap-add method is used to break long signals into smaller segments for easier processing.

In other words, in the frequency domain the FFTs of both signals are multiplied and the result will be converted back into the time domain. For filter kernels longer than about 64 points, FFT convolution is faster than the standard convolution, while producing exactly the same result. The time complexity of using a normal convolution for a matched filter is $O(n^2)$ (where n is the length of the signal in our case), while the complexity for using FFT convolution is $O(n.log.n)$ [23].

We implement the matched filter using FFT convolution. In our solution, the convolution class with FFT is created and initialized using 1024 points. Since the length of generated Gold sequence is 567, the filter Kernel is padded with 457 zeros to bring it to a total length of 1024 points, which is a power of 2 that is greater than the length of the used Gold sequence.

The waveform of the recorded signal and the output of FFT convolution are shown in Fig. 4.3.

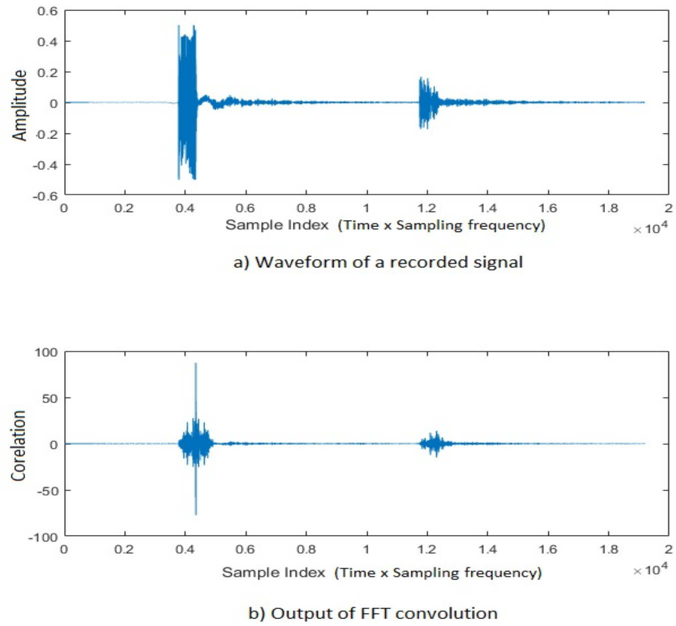


Figure 4.3: Recorded signal and FFT convolution output

4.4 Recording and Audio Latency Issues

If a device runs Signal Capturing (SC) module and Ranging (R) module while multiple other devices are emitting signals to it, it has to record all the received signals simultaneously or at slightly different times. To address this need, we have implemented the recording module to record all the audio signals to a circular buffer with a fixed length. Thus, the device with the SC module can record all the audio signals and can process the recorded signals with specific length from the buffer.

The SC device may have one recording module and several processing modules, which use the recorded data simultaneously. While the recording thread writes the audio signal to the buffer, the buffer is locked by it. As a result, no processing module can access the buffer until the recording thread unlocks it. Moreover, the processing module also locks the buffer when it reads the data with a specified length from the buffer. Therefore, there exist latencies in reading/writing from/to the buffer. However, these latencies are not quantified in any smartphones. The latencies are related to the minimum buffer size of the audio devices. By optimizing the buffer size, we could reduce the latencies. For the new version of the Android phones, the latencies have been reduced, but they still exist. In the future, we will investigate on how to measure the latencies to further improve the ranging accuracy.

Chapter 5

COMMUNICATION PROTOCOL DESIGN

The communication for CDMA code and timestamps and also distance in formations can be done using either Bluetooth or Wi-Fi.

We are using Bluetooth version 4.2 which supports multiple pairs and has high functional range. Depending on type of device BLT version 4.2 coverage range sometimes goes up to 80 meter. As of today the direct Wi-Fi works with only one pair of phone. In the next version direct Wi-Fi is suppose to be able to support multiple pairs. We have developed two version for our application, one which uses BLT for communication and one which uses Wi-Fi for communication. Since direct Wi-Fi does not support multiple pairing we decided to use a router to connect all the phones to each other for now. As soon as new version of direct Wi-Fi would be released teh app can remove the router and switch to use the direct Wi-Fi. Two ports are used to establish the connections and communicate with each other. One port is used for advertising and finding pairs, the other is used for communication. In this project, we call the device, which first uses the SGT module and sends Wi-Fi and acoustic signals, as client. The client peer device is the server. Since servers and clients communicate with each other to exchange messages such as request start, new seed value, synchronizing signals, and timing indices, it is very important to design the communication protocol efficiently. The user datagram protocol (UDP) is suitable for our purpose, where the error checking and correction are either not necessary or will be performed in the application level.

The communication protocol is implemented as shown in Fig. 5.1. Server Side: First, the server broadcasts the $SERVER-HERE$ message to clients to notify them that it is the server. Second, the server receives the request signal from one client and creates the new Seed value (for Gold code) for that client. Third, the server sends the new seed to the client. Then, the server and the client are ready to communicate with each other for ranging. If a new client sends a request to the server, the server generates a new Seed to identify the new

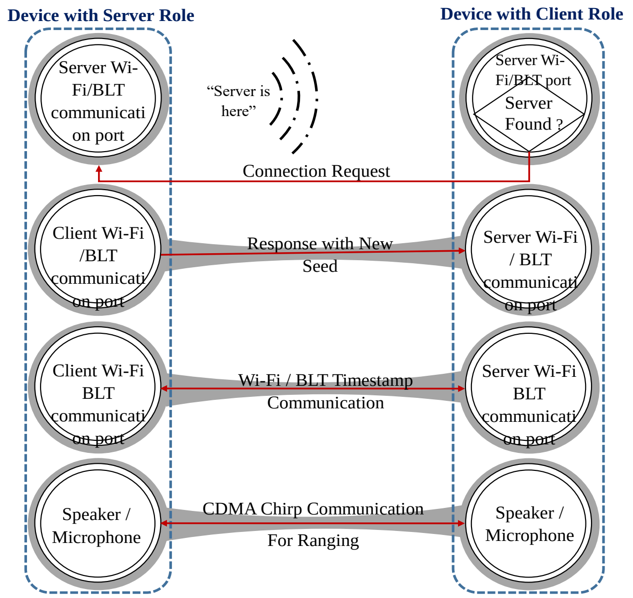


Figure 5.1: Communication Protocol

client. Note that smartphones do not have pre-allocated Gold codes. A server randomly select a code from the 3600 candidate ones (Sec.4.2) for each of its clients. As the number of clients of each server is much less than 3600, the probability of code collision is low even when there exist multiple servers.

Client Side: First, the client finds the servers that broadcast the $SERVER - HERE$ message. Second, the client selects one server from the server list and sends the request signal to the server and receives the response with a new seed value from the server. Third, the client start transmitting the acoustic and Wi-Fi signals with the received seed for ranging. After completing one ranging task, a client can select a new server for another ranging task. To support multiple pairs ranging, a server creates a connection thread for each client.

The server and the client send and receive the message packets using JavaScript Object Notation (JSON) object. They get the *key* and *value* pair and IP address from JSON string. When multiple clients send their signals at the same time, the server identifies the ids of the clients from JSON string and responses to the clients. Note that a client can also select multiple servers and sends its ranging requests in different threads. By this way, we support multiple pairs of smartphones ranging.

Chapter 6
PHONE-PHONE RANGING APPROACHES

In this section, we first introduce the Double-chirp ranging model, which is able to support multiple devices. It eventually yields the accuracy of 5 cm. However, since Double-chirp is a two-way communication model, it has high overhead in terms of run time and it can not provide real time ranging. Therefore, for the cases of mobile scenarios, such as two drones, using Double-chirp for ranging will add an unacceptable error to the ranging result. The effective accuracy for collision avoidance is not acceptable as the system is not providing any real time ranging. To address this problem we also develop a Single-chirp ranging system, which is a one-way communication model. Although it is not as accurate as Double-chirp, but it has effective accuracy for collision avoidance essentially for moving targets as it has much fewer overhead. The accuracy of Single-chirp has been evaluated to be 10-50 cm which still is an acceptable accuracy and higher than many of device-device ranging solutions.

6.1 Double-chirp Ranging

To avoid clock synchronization, we use Double-chirp ranging mechanism to calculate the traveling time of the acoustic signal. The audio signals are played and recorded using Port Audio application programming interfaces (APIs). Note that the matched filter is used to estimate the time when the audio signal is received.

We set the length of the chirp signal as 50 milliseconds while the multi-path effect can be reduced, a level of correlation is gained, and matched filter can be processed faster. We used the Gold code to encode the acoustic chirp signal to support multiple users simultaneously.

As we mentioned in Sec. 5, we call the device, which first sends Wi-Fi/BLT and acoustic signal as client, and the other as server. Both the client and the server have SGT, SC, and R modules in Double-chirp. In Fig. 6.1 the mechanism which we use to implement Double-

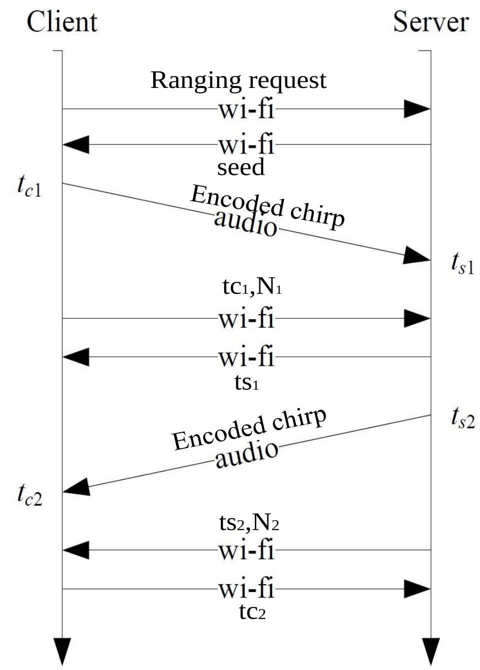


Figure 6.1: Double-chirp ranging mechanism

chirp in our system is shown. The detailed communication transactions are as follows: (i) The client chooses a server from the server list and sends a request signal to the server. When the server receives the request signal, it creates a new seed for the client. The server then sends the new seed to the client. (ii) The client performs the SGT module to generate the acoustic signal using the received seed and play it through the speaker. Note that the length of this signal is fixed. The client records the time, when it plays the audio signal as tc_1 and sends tc_1 to the server through Wi-Fi/BLT. The server performs the SC module to receive the Wi-Fi/BLT signals and record the audio signal through its microphone. Then, it performs the R module to estimate the receiving time ts_1 using the matched filter. (iii) The server plays SGT module to send ts_1 to the client through Wi-Fi/BLT. Then the server generates the acoustic signal, using the seed, plays it through the speaker, and records the time as ts_2. The sever will send ts_2 to the client through Wi-Fi/BLT as well. The client receives the audio signal (SC module) and uses the matched filter to estimate the receiving time tc_2 (R module). (iv), the client sends tc_2 to the server through Wi-Fi/BLT. Now, both client and server collect all the timestamps.

Denote the traveling time of the audio signal from one user to the other by τ. Assume the distance between client and server is fixed within Double-chirp time. The time difference between the server's and the client's clocks is Δt, then, we have the followings:

$$ts_1 = tc_1 + \tau + \Delta t \tag{6.1}$$

$$tc_2 = ts_2 + \tau - \Delta t \tag{6.2}$$

Therefore, without synchronizing the clocks of the two phones, we can calculate the traveling time with the time stamps as follows:

$$\tau = \frac{(ts_1 - tc_1) + (tc_2 - ts_2)}{2} \tag{6.3}$$

6.1.1 Dealing with Uncertainties

The operating system (OS) has a processing delay. It is impossible to record the exact time instance of an event with the required accuracy. In addition, the hardware also has some processing latency in receiving signals. There is a difference between the receiving timestamp, which has been achieved referring the local time, and the actual moment that the signal arrives at the receiver.

This difference causes a significant error in the calculations. In Fig. 6.2 the timing uncertainties between client and server when doing Double-chirp ranging is illustrated. Two timelines are drawn in the figure. The upper timeline illustrates the local time of client, and the bottom one shows the local time of the server. Let tc_1 be the time, when the client orders its speaker to emit the acoustic signal. However, there are some delays in software and hardware, which cause uncertainties in the time of emitting signal. We consider that the actual time, when the speaker physically emits signal, is $\vec{tc_1}$. Note that $\vec{ts_1}$ is the time, when the signal arrives at the server's microphone. Similarly, due to the receiving time uncertainty, the server will consider ts_1 as the time when it receives the acoustic signal.

Similarly, when the server plays the acoustic signal back, we also have send and receive time uncertainties. We denote $\vec{ts_2}$ and ts_2 as the times, when the server orders to send

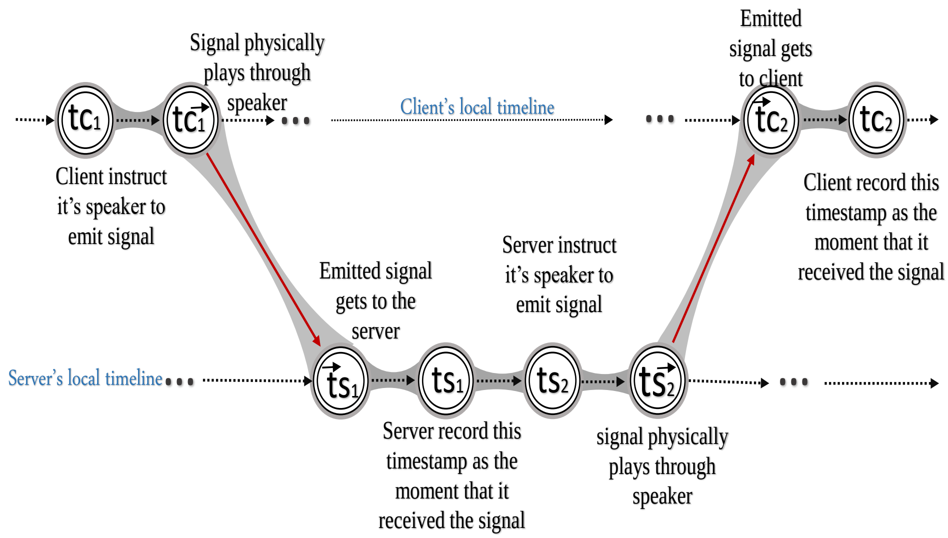


Figure 6.2: Timing uncertainties between client and server in Double-chirp ranging

out a sound signal and when the signal is physically sent out, respectively. $\overrightarrow{tc}_2$ and tc_2 are the times, when the signal from the server arrives at the client's microphone and when the client considers that it receives the signal, respectively.

These uncertainties, in other words, are the time differences between the moments, when the OS instructs its speaker to emit the sound signal and the actual time when the speaker physically emits sound, and between the moments, when the microphone gets the signal and when the system thinks the microphone gets the signal. These latencies are also related to the minimum buffer size of the audio playing module supported by the device.

6.1.2 Sample Counting

As the recorded signal is always sampled at a fixed frequency, instead of using timestamps, we decide to use sample counting [26, 34] to ignore the uncertainties. We count the number of samples in the recorded file. In this ranging scheme, the local clock of the device is never used. We can rewrite equation (6.3) as:

$$\tau = \frac{(ts_1 - ts_2) + (tc_2 - tc_1)}{2} \tag{6.4}$$

Using multi-threading, we run thread 2 (record audio) on each device. When a device sends the audio signal, since the speaker and microphone of the devices are close to each other, we assume that the device's microphone receives the audio signal as soon as its speaker plays it. The same thread continues recording to get the audio signal from the other device. The number of samples between the first sample of the audio signal and the first recorded sample can be estimated via the matched filter. Let it be N. Since both the sample rate of playing the audio signal and that of recording are 48 kHz, the time difference between the receiving audio signal and playing audio signal is $N/48000$. We denote the number of samples between two audio signals in the client and the server as N_1 and N_2, respectively. We can derive (6.4) as

$$\tau = \frac{(N_1 - N_2)}{2 \times 48000} \tag{6.5}$$

6.2 Single-chirp Ranging

The basic idea of Single-chirp ranging is to synchronize the smartphones' clocks via multiple Wi-Fi packets. Then, the distance between two smartphones can be estimated by sending only one acoustic signal. In this subsection, we will describe our technique for clock synchronization. Then, we will introduce our design of the Single-chirp ranging system.

6.2.1 Clock Synchronization Protocol

Clock synchronization cannot be simply done by sending the time of the server to the client through Wi-Fi/BLT due to the uncertainties in Wi-Fi/BLT transmissions and system processing. There has been little work in addressing the Wi-Fi/BLT's send and receive time uncertainties at the software level. Most previous studies were focused on minimizing the uncertainties by customizing hardware design so that the system can precisely control and obtain the exact instant when a signal is sent or received [9]. These solutions are, however, not applicable for our system.

In Fig. 6.3, the timing relation between events when the server and the client are sending/receiving Wi-Fi/BLT packets to/from each other, is illustrated. The Δ terms for each

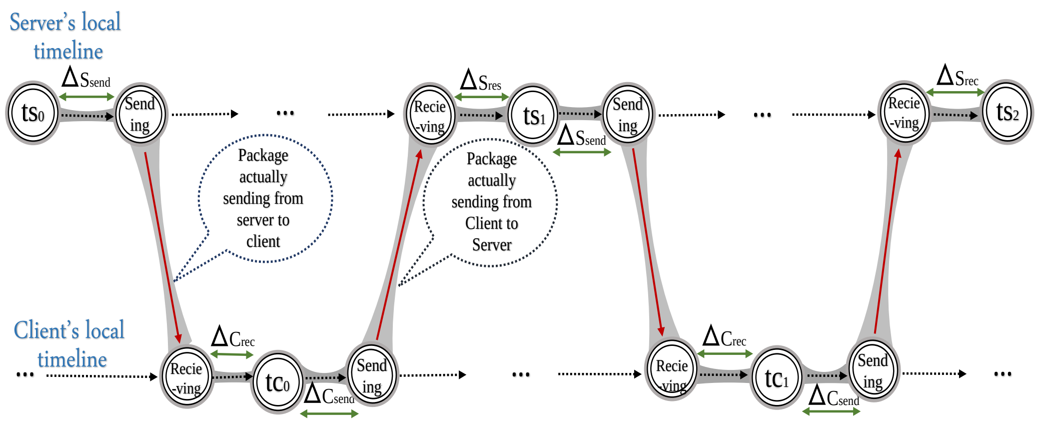


Figure 6.3: Timing relation between events when two devices are sending Wi-Fi/BLT packets.

transmission are the uncertainties. These uncertainties can be as large as 100 ms [26]. From Fig. 6.3, we can see that Δs_{send} is the time difference between the moment when the server issues a packet to be sent and the moment when the packet is actually sent out. $\Delta s_{receive}$ is the time difference between the moment when the server actually receives the packet and the moment when the packet is delivered to our App. Δc_{send} and $\Delta c_{receive}$ are similarly defined in the client side.

Though the Δ terms vary, we assume that they are almost the same within two or three continuous transmissions. In Fig. 6.3, the server first gets local time (ts_0) and sends it to the client. The client receives the packet containing ts_0 via Wi-Fi/BLT at its local time tc_0. The client sends tc_0 to the server immediately. Then, the server receives the packet with tc_0 and gets the timestamp ts_1. The server immediately sends ts_1 to the client. Then, the clients receives ts_1 and gets timestamp tc_1. The client, then, immediately sends tc_1 to the server.

Assuming that the clock difference between the client and the server is $T_d = T_c - T_s$, where T_c and T_s are the clocks of the client and the server at the same instance, respectively. From the first three transmissions we have:

$$tc_0 = ts_0 + T_d + (\Delta S_{send} + \Delta C_{receive}) \tag{6.6}$$

$$ts_1 = tc_0 - T_d + (\Delta C_{send} + \Delta S_{receive}) \tag{6.7}$$

$$tc_1 = ts_1 + T_d + (\Delta S_{send} + \Delta C_{receive}) \tag{6.8}$$

Assuming that the two smartphones are equivalent in terms of computing capability, communication capability, traffic demand, and background noise, and that the equation $\Delta S_{send} + \Delta C_{recieve} = \Delta S_{recieve} + \Delta C_{send}$ holds for three continuous transmissions, we have

$$T_d = \frac{2tc_0 - (ts_0 + ts_1)}{2} \tag{6.9}$$

$$T_d = \frac{-2ts_1 + (tc_0 + tc_1)}{2} \tag{6.10}$$

By taking the average of (6.9) and (6.10), we have

$$T_d = \frac{2tc_0 - 2ts_1 + (tc_0 + tc_1) - (ts_0 + ts_1)}{4} \tag{6.11}$$

Then, we use (6.11) to estimate T_d. Note that we can get a T_d from every three continuous Wi-Fi/BLT transmissions. To find the best estimation of T_d, We design a test, which has 200 continuous transmissions between two smartphones. After running the test on several smartphone-pairs, we have the results as shown in Fig. 6.4. The ground truth of the time drift is obtained through connecting both of the smartphones to a PC through a USB cable and getting their local times to calculate the difference [30].

As shown in Fig. 6.4, when the number of packets is 20, we get the best result. The accuracy of estimation drops significantly when the number of packets increases because of the congestion and the large transmission and receiving buffer delays. Hence, the best case is to calculate the clock skew based on the packet number 18, 19, 20. It means that we only need to send 20 packets in our solution.

6.2.2 Single-chirp Ranging Approach

In Single-chirp ranging, distance estimation is done by jointly utilizing the clock synchronization protocol and the previously mentioned techniques. First, a smartphone (server)

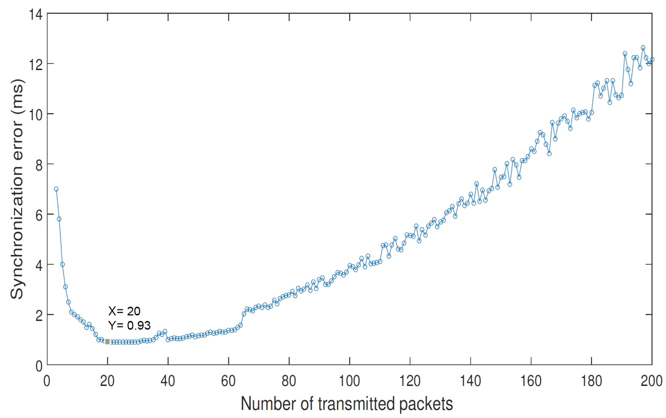


Figure 6.4: Synchronization error over the number of packets communicated between two devices

sends a '$ServerHere'$ message. Another smartphone (client) connects to the server. The server and the client run the clock synchronization protocol to calculate the clock difference. Then, the client performs the SGT module to generate and transmit its acoustic packet with timestamps (t_c). The client records this acoustic packet through its own microphone and uses the matched filter to find the exact time, $t_c^{\rightarrow}$, when the acoustic packet was received by itself. t_c and $t_c^{\rightarrow}$ are different due to the system timing uncertainties of recording and playing acoustic packets. We consider $\Delta t = t_c^{\rightarrow} - t_c$ as the processing delay. The client sends Δt to the server. The server performs SC module to receive the Wi-Fi/BLT signals and record the audio packet. The server, then, performs the R module to obtain the receiving time (t_s) via using the matched filter. The flying time of the acoustic signal, therefore, can be calculated as

$$\tau = t_s - t_c - \Delta t \tag{6.12}$$

6.3 Server-less Localization

Above, we discussed how to measure distances between two devices or multiple devices using acoustics.Now we explain how to use these ranging information to localize several devices simultaneously relative to each other.

6.4 On-Demand Fast Localization

The localization starts from multi-coordinate systems, and the position transformation is integrated with LBS applications information transmissions. In particular, ODFL captures the earliest time for starting LBS applications. Each intermediate user on the route can help transforming the information of the position of the source to the one in its next-hop users coordinate system until the destination successfully obtains the position of the source in its own coordinate system. Based on this per hop location transformation method, LBS applications can start right after the information source localizing itself in its own coordinate system. In other words, information transmissions of LBS applications can start before all the users are localized in the same coordinate system. The start of the second procedure in the original pipeline no longer depends on the success of the first procedure.

6.5 Localization Framework

We devise a framework of ODFL which employs coordinate system transformation on a per hop basis, which consists of three components: (1) LCS construction; (2) embedded information selection; and (3) position transformation. 1) LCS construction: The first component of ODFL is to construct a local coordinate system at each user on the information route so that the bridge users can be localized. The procedure of LCS construction is composed of two steps: (1) local information collection and (2) LCS construction. In Step (1), each user on the routing path first broadcasts an information-request message to its direct neighbors. Each of the users neighbors replies with the information of its one-hop topology, which can be obtained via round trip communications for neighbor discovery. For the users on the information transmission route, they collect information in a chain procedure. In other words, each intermediate user initiates its information collection only after receiving the previous users information request message. Each intermediate user can derive its two-hop topology for LCS construction as shown in Fig. 6.5. Note that there are no edges between two-hop neighbors. An LCS can be constructed by finding three mutually connected users (such as users a, b, and X) to form a coordinate system, which is denoted by its LCS ID (such as $\langle a, b, X \rangle$).

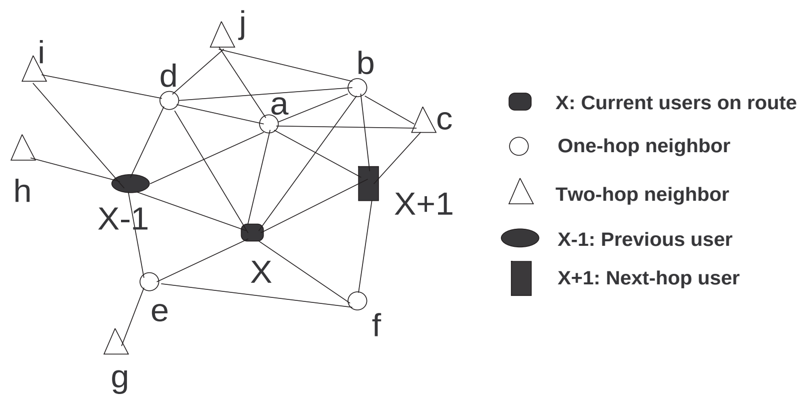


Figure 6.5: An example of user's two-hop topology for LCS construction

Multiple LCSs may exist in the two-hop topology of the user. The set of users that can be localized in each LCS can be obtained via trilateration in $O(\beta^6)$ time, where β is the maximal node degree in the users two-hop topology. Then, these LCSs can be merged into a number of candidate LCSs (denoted by CLCSs) according to the condition of coordinate transformation. Therefore, the current user in Fig. 6.5 can have a few CLCSs and the users that can be localized in these CLCSs after Step (2). Note that some users may be localized in more than one CLCSs, and that any two CLCSs cannot localize more than 2 common users as they are not mutually transformable. For the example in Fig. 6.5, the current user can have five CLCSs: $\langle a, b, X \rangle$, $\langle d, i, X-1 \rangle$, $\langle e, X-1, X \rangle$, $\langle e, f, X \rangle$, and $\langle f, X, X+1 \rangle$. Users $b, c, d, j, X-1, X$, and $X+1$ are local CLCS $\langle a, b, X \rangle$; users d, i, and $X-1$ are localized in $\langle d, i, X-1 \rangle$; users $e, X-1$, and X are localized in $\langle e, X-1, X \rangle$; users e, f, and X are localized in $\langle e, f, X \rangle$; users f, X, and $X+1$ are localized in $\langle f, X, X+1 \rangle$.

2) Embedded information selection: The objective of this component is to embed appropriate information into the message at each hop to transform source position on a per-hop basis. Suppose that the current users can obtain the corresponding users of the source in their CLCSs. They embed the information as shown in table 6.1 into the message for helping the next-hop user with position transformation. Based on the embedded information, the next-hop user can transform the source positions to the corresponding ones in its CLCSs

when three or more bridge users in a row of table 6.1 can be localized in one of its CLCSs. Note that the next-hop user may obtain the sources corresponding positions in multiple CLCSs.

Table 6.1: Table of CLCSs information which will be embedded as a massage at each hop

Source Position in CLSCS1	CLCS1 ID	Bridge user's positions in CLCS1
Source Position in CLSCS2	CLCS2 ID	Bridge user's positions in CLCS2
Source Position in CLSCS3	CLCS3 ID	Bridge user's positions in CLCS3
....	....	......

In general, each user on the information transmission route is ready for the position transformation after LCS is constructed. As there is no edge between any 2 two hop neighbors, any of the current users two-hop neighbors may also appear in the next-hop users two-hop topology. Moreover, the current user cannot identify the users that cannot be localized by the next-hop user because it does not know the next-hop users two-hop topology. As a result, every localized user could be a bridge user. In order to maximize the probability of position transformation, we conservatively choose to embed all the users positions in the CLCSs, where the source and at least three non-collinear bridge users can be localized, into the message.

3) Position transformation: Upon receiving the embedded information from the previous user, the current user transforms the position of the source to the corresponding one in its CLCS through solving (x, y) from Eqs.6.13, where (x, y) and (x', y') are the coordinates of the source in the current users CLCS and in the previous users CLCS, respectively; (x_1, y_1), (x_2, y_2), and (x_3, y_3) are the positions of the three bridge users in the current users CLCS; $(x'_1, y'_1), (x'_2, y'_2)$, and (x'_3, y'_3) are the bridge users corresponding positions in the previous users CLCS; Note that x and y are the only unknowns in Eqs 6.13 as the current user can obtain (x', y'), (x'_1, y'_1), (x'_2, y'_2), and (x'_3, y'_3) from the previous users embedded information, and it can also acquire $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3) after its LCS construction procedure.

The solution is unique as long as the three bridge users are non-collinear.

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = (x' - x'_1)^2 + (y' - y'_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 = (x' - x'_2)^2 + (y' - y'_2)^2 \\ (x - x_3)^2 + (y - y_3)^2 = (x' - x'_3)^2 + (y' - y'_3)^2 \end{cases} \tag{6.13}$$

Chapter 7
EXPERIMENTAL EVALUATION

The most important factor for evaluating a range-base localization system is ranging accuracy. In this section, we first evaluate the ranging accuracy of both Double-chirp and Single-chirp approaches then we will evaluate the final results of positioning system.

7.1 Experimental Setup

Our system has been evaluated in both indoor and outdoor environments. The indoor environment is a room approximately 40 m x 60 m with some tables in the middle and an ambient environmental temperature of 25° C. Outdoor environment is an open area beside a street, surrounded by tall buildings and automobiles with ambient temperature of 9° C. The system has been deployed on 8 Android smartphones, Samsung Galaxy S7 (Android 7.0), Motorola Nexus 6 (Android 7.0), LG Nexus 5 (Android 4.4.1), Samsung Galaxy Nexus (Android 4.3), Samsung S4 mini (Android 5.1.1), Google Pixel (Android 6.1), Huawei Mate 10(Android 8.0), and Google Nexus 6 (Android 7.1). We use them to evaluate the Double-chirp and the Single-chirp in four different scenarios:

- Scenario 1: Indoor environment and using just one pair of phones for ranging.

- Scenario 2: Indoor environment and using multiple pairs, where all the phones run ranging algorithm at the same time. The Gold CDMA acoustic signals are played in a way to have overlaps.

- Scenario 3: Outdoor environment using one pair of phones for ranging.

- Scenario 4: Same as Scenario 2 but is performed in an outdoor environment.

In Fig. 7.1, the experimental environments and a running example of our App are shown. The App example is for an indoor environment where the actual distance between two smartphones is 10 m.

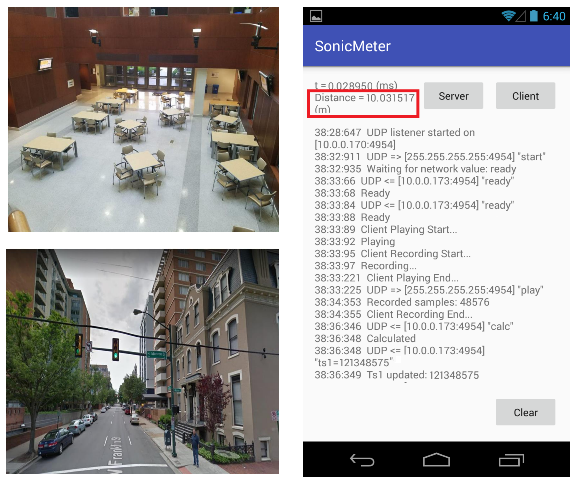


Figure 7.1: Experimental environments and App running example

7.2 *Double-chirp Results*

In Fig. 7.2, the mean of estimated distance is plotted over the real distance using Double-chirp for all the four scenarios. The distance between two devices has varied from 15 cm to 60 m in the indoor scenarios. In the outdoor scenarios, due to a high level of the noise, some of the phones with a weaker speaker (LG Nexus 5 and Samsung S4 mini) will fail when the distance is 50-55 or 11 meter. Therefore, we consider the effective range of the outdoor ranging as 50 meters. At each distance, the Double-chirp ranging is run 20 times. We obtain the mean of the measured distance and estimation error over 20 instances. There was no object between two phones when the distance is between 15cm to 3 meters. Using Double-chirp we can achieve 5 cm precision. The run time of Double-chirp varies between 1 and 2 seconds.

In the case of multiple pairs, the overhead increases with the increase of the number of users while the accuracy does not change much in our experiments as shown in Table 7.1.

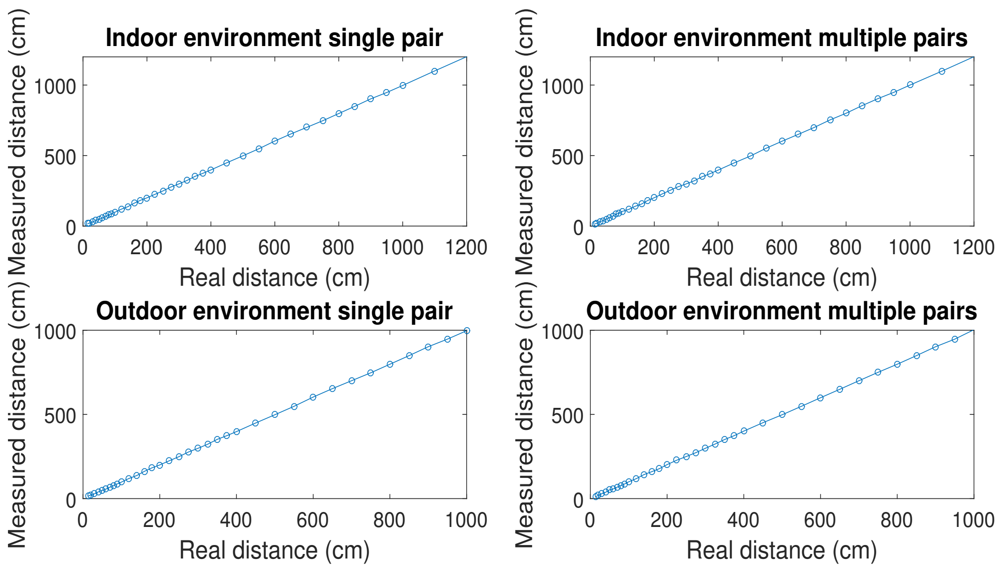


Figure 7.2: Estimated distance vs. real distance using Double-chirp

Note that the accuracy and the run time may vary depending on the types of the phones, and the quality and the power of their speakers and microphones. Due to the page limit, we will report more detailed results in appendix.

7.3 Single-chirp Results

In Fig. 7.3, the mean of the estimated distance is plotted over the real distance for all the four scenarios while using Single-chirp.

The distance of the devices varies from 50 cm to 60 cm in all the scenarios. In Single-chirp, since only one acoustic signal is transmitted, its chance of getting polluted with noise and failing to reach the receiver is only half of that for Double-chirp. Therefore, we reach a higher effective range. At each distance, the App is run 20 times to get the mean of measured distance and estimation error.

Single-chirp has the run time of less than 0.5 seconds. Though Single-chirp has a fast response time, it has a lower accuracy compared with Double-chirp. Single-chirp gives us the accuracy of 10 cm to 50 cm based on results from the best device pair. Fig. 7.4 shows the estimation error at each distance for both of Double-chirp and Single-chirp approaches

Table 7.1: Absolute ranging errors / their standard deviations of different Android devices by using Double-chirp in the indoor environment for simultaneous ranging among multiple pairs of phones

Range (cm)	Sam S7	Moto Nexus	Sam Nexus	LG Nexus	Sam S4m
100	3/1.4	4/1.6	2/0.9	2/1.8	1/1.2
375	3/1.1	3/0.8	3/0.7	3/1.5	2/2.1
550	3/0.5	2/1.6	2/2.3	2/2.5	3/1
750	4/0.7	3/1.7	2/2.5	2/2.2	3/2
950	2/1	1/1.1	3/2	3/2.6	5/1.4
1200	4/1.4	1/1.5	2/2.2	2/2	3/2.1

in indoor environment while performing Multi-pair ranging.

7.4 Error Analysis and Accuracy

We have performed ranging on 6 different Android devices using both Double-chirp and Single-chirp in all the scenarios. We have demonstrate the ranging error of Samsung Galaxy S7 at several different distances using both Double-chirp and Single-chirp in Scenarios 2 and 4 in Fig. 7.5. It is clear from Fig. 7.5 that Double-chirp is a more accurate method in both indoor and outdoor environments, and its ranging errors in both indoor and outdoor environments are almost the same (the error is slightly larger in the outdoor environments especially at longer distances). The effective range of our system while using Double-chirp in indoor environments is around 60 m. In outdoor environments, it is around 50 m. The reason is that with the increase of distance in outdoor environments, the SNR decreases and the beeps more likely get polluted by noise. As a result, they cannot be detected at the receiver side.

One of the limitations of two way acoustic ranging methods is their small effective range (4 m)[26] in indoor environments due to the multi-path effect. By designing a different acoustic signal, we are able to mitigate the multi-path effect and achieve a higher effective

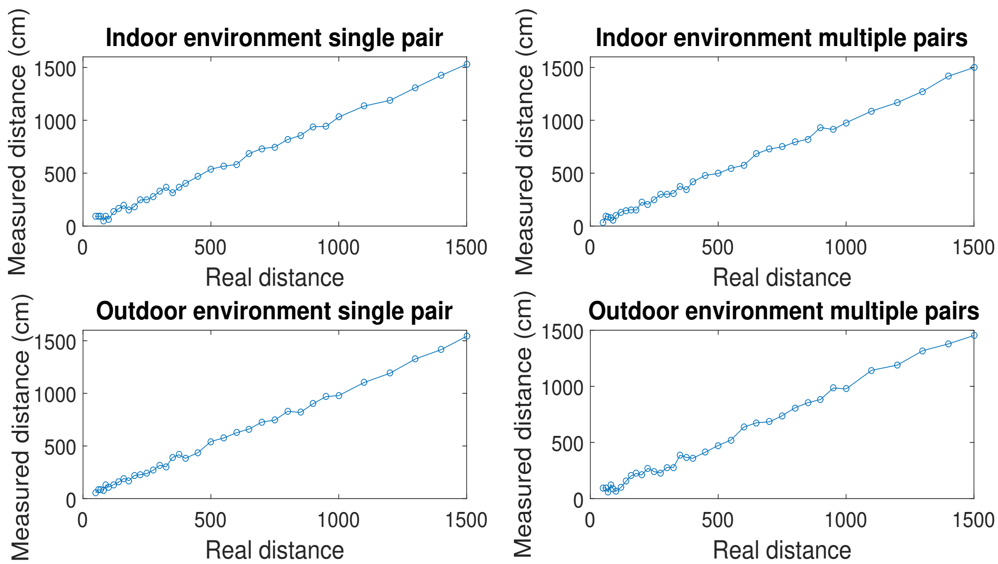
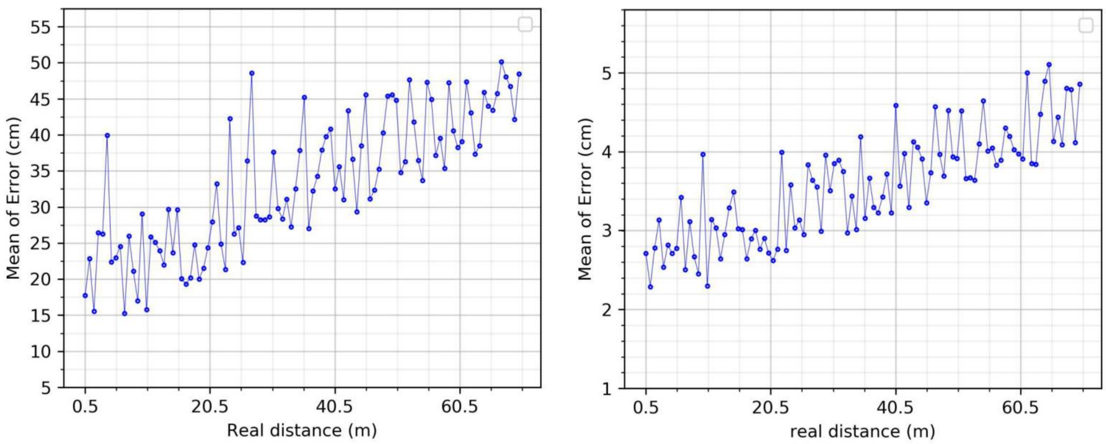


Figure 7.3: Estimated distance vs. real distance using Single-chirp

range. The effective range of our ranging system is limited to the power of the speaker on each device.

Single-chirp has a larger error compared to Double-chirp. The major source of the error is the uncertainties in both client and server. To mitigate these uncertainties, as we explained in Sec.6.2.2 the client performs self-recording when emitting its acoustic signal. It can measure the uncertainties within its calculations. Finally, this calculated uncertainties will be reduced from the final signal flying time. Note that, in this solution, we are assuming that recording uncertainties of the client is equal to recording uncertainties of the server. This assumption is a source of error itself, because in reality the uncertainties are not the same. Another source of error, as we can see in Fig. 10, is the time synchronization error. Even after applying the synchronization technique, the pair of devices still have a slight clock skew. In Single-chirp, since we use only one signal, the chance of losing the signal is less than Double-chirp. Therefore, the effective range of Single-chirp is higher than Double-chirp in both indoor and outdoor environments.

We have developed Double-chirp as a fast two way acoustic ranging method. The highest overhead of the system is in the detection part. To reduce this overhead, as mentioned



Single-Chirp Error   Double-Chirp Error

Figure 7.4: Mean of error vs. real distance using both ranging approaches

before we use FFT in our matched filter. We also remove the self-recording task of two in our Double-chirp in both devices at the beginning of ranging. As a result, the number of times of using matched filter has been reduced by two. In addition, the time periods, of generating, playing, and recording the signal is removed from both devices as well. However, in the Double-chirp, the system still has a high overhead (1-2 seconds). As the number of the users increases, it will get worse. On the other hand, the overhead of Single-chirp is at least two times less than Double-chirp. Overall, the processing time will be limited by the phones' processing power. In addition, when more users participate in the system the server will open more threads to support multiple pairs ranging. Though there is no limitation on the number of threads, each of these threads has computational and communication costs, which cause some latency's in the system processing.

For the audio waves, the propagation speed depends on the properties of the substance through which the wave is traveling. The speed of the sound has been calculated as $V_s = 331.3 + 0.606\theta$ m/s [32] (θ is the air temperature in Celsius). This model is for the acoustic wave propagation in dry air. It may be another source of error. Note that, in these tests, the phones were placed face to face and there was no object in between to block the line

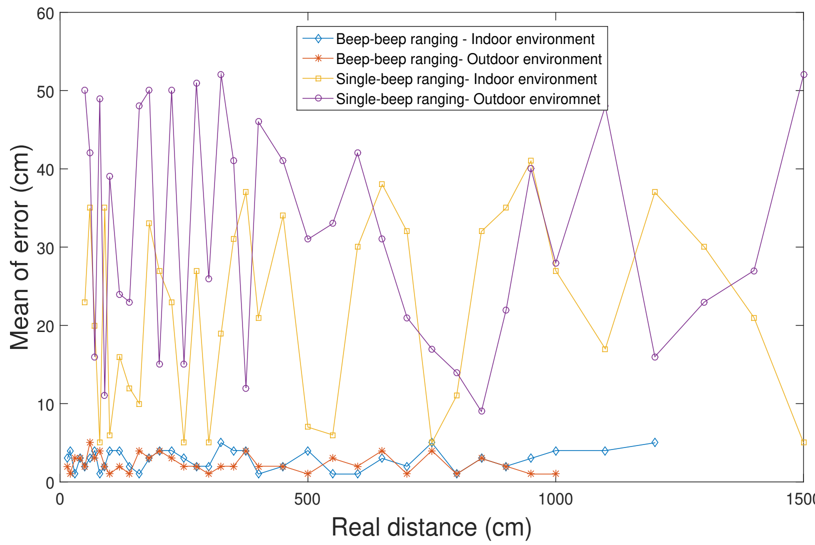
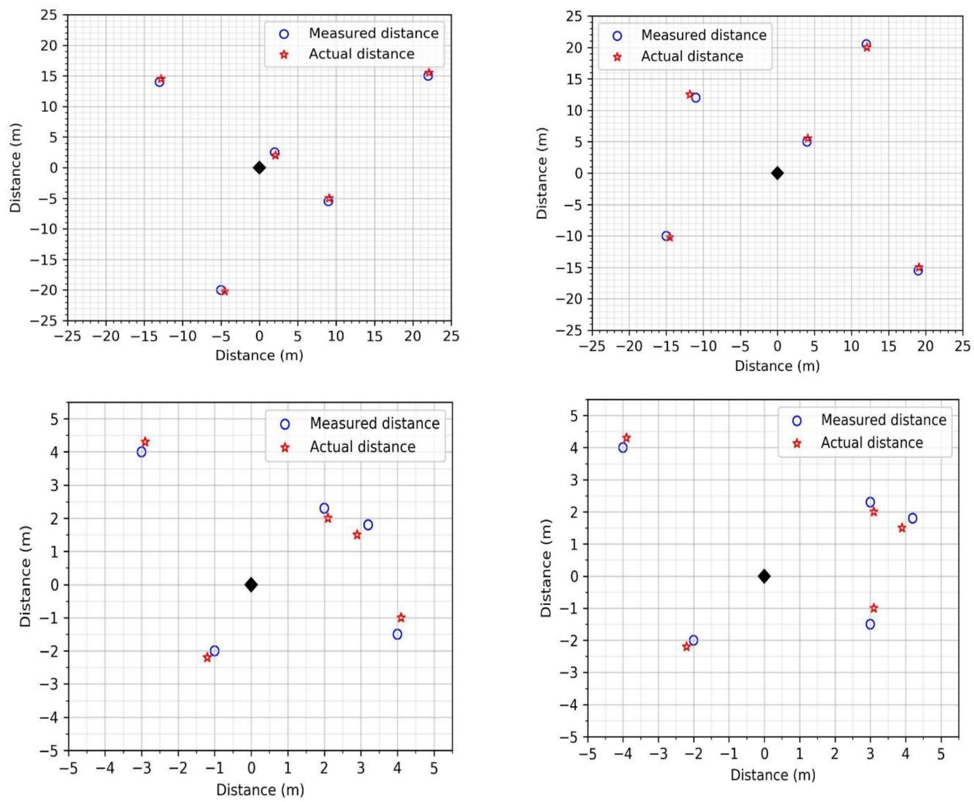


Figure 7.5: Ranging errors for Samsung Galaxy S7 in Scenarios 2 and 4

of sight. We also performed some additional experiments to our system. when the phones are not face to face but they still are line of sight, depending on the orientation of the speaker and microphones on the phones, the error will increase up to several centimeters. Moreover, when the phones are not line of sight and there are obstacles such as people or furniture introduced at different range, we observed that the magnitude of error will increase up to several meters for outdoor environment and up to a meter for indoor environment (there will be a second path for the acoustic signal due to reflection of the walls and the ceiling). Once the number and the size of obstacles increases, they absorb a high proportion of transmitted signal energy, the match filter can't detect the signal. An intermediate way of dealing with this problem could be assume that there is a layer above the system, based on existing approaches to none line of sight identification and mitigation, which could be used to distinguish between line of sight and none line of sight measurements and only use the line of sight ones. In future work we will challenge proposing an accurate estimation method for none line of sight ranging.



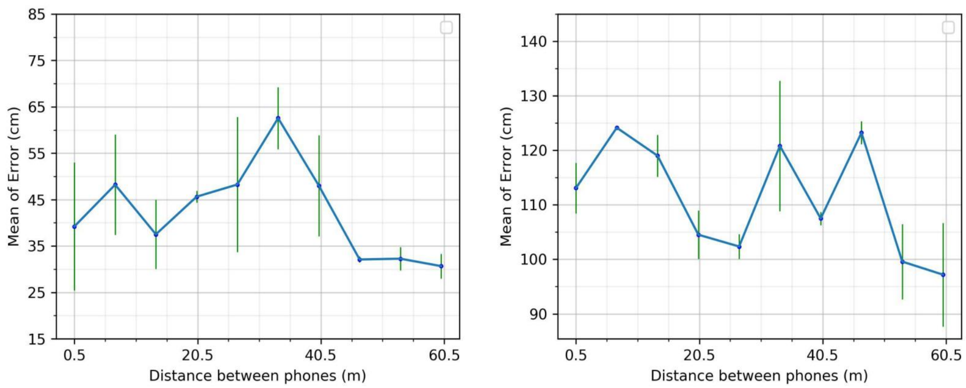
Output of application ran in indoor environment

Output of application ran in outdoor environment

Figure 7.6: Application output ran in indoor and outdoor environment

7.5 Localization Experimental results

Figure 7.6 shows some example of application output ran to locate 6 phones in indoor and outdoor environment. In each of the figures as you see the users device is in the center the other devices has been located relative to the user. The red stars shows the real position of each phone and the blue circle shows the estimated position. We have used Double-Chirp for ranging. The distance between devices has been increased up to 60 meter. In indoor environment positioning accuracy were around 70-80 cm with the confidence of 20 cm. in outdoor environment we could achieve accuracy of 140 cm with almost 45 cm confidence.



Indoor environment

Outdoor environment

Figure 7.7: Mean of error over distance for positioning in indoor and outdoor environment

at each distance the we ran the 10 application times. Fig. 7.7 shows mean of estimated position error over the distance for indoor and outdoor environment.

Chapter 8

CONCLUSIONS

In this thesis report, we have designed, implemented, and evaluated our fast multi-pair smartphone localization systems. The are purely software-based solutions, which do not require any additional hardware or infrastructure support. They can provide 100 cm positioning accuracy in indoor environment and 130cm positioning accuracy for outdoor environments, where the range is up to 60 m in both indoor and outdoor. The system is dynamic, can support multiple devices and multiple pairs, new devices can join the system without any interference problem.

We are encouraged by our experiences with the proposed localization systems, which are simple, effective, and cost-efficient. In the future, we will further improve the signal detection component and analyze the processing delay in Single-chirp so that we can provide higher ranging accuracy in less time which will lead to faster and more accurate positioning.

BIBLIOGRAPHY

[1] Isaac Amundson and Xenofon D Koutsoukos. A survey on localization for mobile wireless sensor networks. In *Mobile entity localization and tracking in GPS-less environnments*, pages 235–254. Springer, 2009.

[2] Pete Steggles Andy Ward Paul Webster Andy Harter, Andy Hopper. Bat System. http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/, 2008. [Online; accessed 4/4/2018].

[3] Gaetano Borriello, Alan Liu, Tony Offer, Christopher Palistrant, and Richard Sharp. Walrus: wireless acoustic location with room-level resolution using ultrasound. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 191–203. ACM, 2005.

[4] Joyraj Chakraborty, Geoffrey Ottoy, Maxim Gelaude, Jean-Pierre Goemaere, and Lieven De Strycker. Acoustic localization of unknown sources with wireless sensor nodes. In *Computer and Information Technology (ICCIT), 2014 17th International Conference on*, pages 488–493. IEEE, 2014.

[5] Wei Cheng, Jindan Zhu, Prasant Mohapatra, and Jie Wang. Time and energy effiecient localization. In *The 11th IEEE International Conference on Sensing, Communication, and Networking (SECON 2014).*

[6] Charles Cook. *Radar signals: An introduction to theory and application.* Elsevier, 2012.

[7] Esmael H Dinan and Bijan Jabbari. Spreading codes for direct sequence cdma and wideband cdma cellular networks. *IEEE communications magazine*, 36(9):48–54, 1998.

[8] Viktor Erdélyi, Trung-Kien Le, Bobby Bhattacharjee, Peter Druschel, and Nobutaka Ono. Sonoloc: Scalable positioning of commodity mobile devices. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '18, pages 136–149, New York, NY, USA, 2018. ACM.

[9] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2/3):187–197, 2002.

[10] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In *International Conference on Ubiquitous Computing*, pages 264–280. Springer, 2002.

[11] Mike Hazas and Andy Ward. A high performance privacy-oriented location system. In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 216–223. IEEE, 2003.

[12] Stephan Hengstler, Dayalan P Kasilingam, and Antonio H Costa. A novel chirp modulation spread spectrum technique for multiple access. In *Spread Spectrum Techniques and Applications, 2002 IEEE Seventh International Symposium on*, volume 1, pages 73–77. IEEE, 2002.

[13] Fabian Höflinger, Rui Zhang, Joachim Hoppe, Amir Bannoura, Leonhard M Reindl, Johannes Wendeberg, Manuel Bührer, and Christian Schindelhauer. Acoustic self-calibrating system for indoor smartphone tracking (assist). In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–9. IEEE, 2012.

[14] Hakan Koyuncu and Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.

[15] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112. ACM, 2012.

[16] Pei H Leong, Thushara D Abhayapala, and Tharaka A Lamahewa. Multiple target localization using wideband echo chirp signals. *IEEE Transactions on Signal Processing*, 61(16):4077–4089, 2013.

[17] Kaikai Liu, Xinxin Liu, Lulu Xie, and Xiaolin Li. Towards accurate acoustic localization on a smartphone. In *INFOCOM, 2013 Proceedings IEEE*, pages 495–499. IEEE, 2013.

[18] Atri Mandal, Cristina V Lopes, Tony Givargis, Amir Haghighat, Raja Jurdak, and Pierre Baldi. Beep: 3d indoor positioning using audible sound. In *Consumer communications and networking conference, 2005. CCNC. 2005 Second IEEE*, pages 348–353. IEEE, 2005.

[19] Miklós Maróti, Péter Völgyesi, Sebestyén Dóra, Branislav Kusỳ, András Nádas, Ákos Lédeczi, György Balogh, and Károly Molnár. Radio interferometric geolocation. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12. ACM, 2005.

[20] mathwork. Gold Sequence Generator. www.mathworks.com/help/comm/ref/goldsequencegenerator.html, 2006.

[21] Rainer Mautz. Indoor positioning technologies. 2012.

[22] Henri J Nussbaumer. *Fast Fourier transform and convolution algorithms*, volume 2. Springer Science & Business Media, 2012.

[23] Haldun M Ozaktas, Orhan Arikan, M Alper Kutay, and Gozde Bozdagt. Digital computation of the fractional fourier transform. *IEEE Transactions on signal processing*, 44(9):2141–2150, 1996.

[24] Kaveh Pahlavan, Xinrong Li, Mika Ylianttila, Ranvir Chana, and Matti Latva-aho. An overview of wireless indoor geolocation techniques and systems. In *Mobile and Wireless Communications Networks*, pages 1–13. Springer, 2000.

[25] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14. ACM, 2007.

[26] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: A high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 1–14, New York, NY, USA, 2007. ACM.

[27] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[28] John G Proakis. *Companders*. Wiley Online Library, 2001.

[29] Zheng Sun, Rick Farley, Telis Kaleas, Judy Ellis, and Kiran Chikkappa. Cortina: Collaborative context-aware indoor positioning employing rss and rtof techniques. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 340–343. IEEE, 2011.

[30] Chris Svensrud. Extending USB to Time-Critical Applications. hhttps://www.sensorsmag.com/embedded/extending-usb-to-time-critical-applications, Mar, 2006.

[31] Harry L. Van Trees, Kristine L. Bell, and Zhi Tian. *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Filtering Theory*. Wiley, 2 edition, 2013.

[32] George SK Wong and Tony FW Embleton. Variation of the speed of sound in air with humidity and temperature. *The Journal of the Acoustical Society of America*, 77(5):1710–1712, 1985.

[33] Bin Xu, Ran Yu, Guodong Sun, and Zheng Yang. Whistle: Synchronization-free tdoa for localization. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 760–769. IEEE, 2011.

[34] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P Martin. Sensing driver phone use with acoustic ranging through car speakers. *IEEE Transactions on Mobile Computing*, 11(9):1426–1440, 2012.

[35] Moustafa Youssef, Adel Youssef, Chuck Rieger, Udaya Shankar, and Ashok Agrawala. Pinpoint: An asynchronous time-based location determination system. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 165–176. ACM, 2006.