An Investigation into the Use of Inexpensive Audio Equipment as a Method of Real-time 3D Sound Source Localization

by

LI,Chenfeng

31/5/2011

Thesis Submitted to

The Hong Kong University of Science and Technology

in Partial Fulfillment of the Requirements for

the Degree of Master of Philosophy

in Computer Science and Engineering

May 2011, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

LI CHENFENG

31 May 2011

An Investigation into the Use of Inexpensive Audio Equipment as a Method of Real-time 3D Sound Source Localization

by

LI,Chenfeng

This is to certify that I have examined the above M.Phil. thesis

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by

the thesis examination committee have been made.

DR. DAVID ROSSITER, THESIS SUPERVISOR

PROF. MOUNIR HAMDI, HEAD OF DEPARTMENT

Department of Computer Science and Engineering

31 May 2011

ACKNOWLEDGMENTS

Special thanks to my supervisor Dr. David ROSSITER, for his patience and actively participation which guided me here.

Thanks to Tony REN who lent me his notebook to record the experiment data.

Thanks to Barren BAI who lent me his digital camera to shoot the photos in the thesis.

Thanks to all my friends who supported me over the past two years.

Thanks to my parents for their great support.

Thank you!

LI Chenfeng

The Hong Kong University of Science and Technology

May 2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

An Investigation into the Use of Inexpensive Audio Equipment as a Method of Real-time 3D Sound Source Localization

by

LI,Chenfeng

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Sound source localization has many applications in a wide range of areas. Time Delay Estimation (TDE) has been the main approach used to solve this problem. However, current research and commercial products are either inaccurate or too expensive. The fact is, the more accurate the device the higher the price. Our research is focused on the key problem behind this dilemma: the resolution problem. We are trying to use devices designed for vocal voice recording to solve the sound localization problem. We have to achieve high precision based on the limited precision of the devices through estimation algorithms. We set our goal to obtain the same results as a more expensive commercial product, but at one tenth of the price. Our experiments show that we are very close to that goal. Another stream of our research was a trial to develop a complete and inexpensive real-time 3D localization system based on energy propagation. However, that system did not end up satisfying expectations. Therefore, we conclude that, it is very likely that the TDE resolution on inexpensive devices can reach the level which is usually required for expensive devices; while the energy propagation based approach is not a good path for further exploration.

CHAPTER 1

INTRODUCTION

1.1. THESIS MOTIVATION AND OBJECTIVE

Controlling a position in 3D space has various usages in oceanic and space engineering, medical surgery, biological experimentation, CAD, computer games, the military, and the like.

There are two main types of applications. The first one is controlling a physical device to carry out work which is extremely difficult for humans to accomplish directly, such as moving a rock on Mars, disarming a bomb, or even simply scratching one's own back. The second is controlling a virtual device, such as a computer game, virtual reality and CAD.

The techniques for controlling a 3D position can be divided into five categories in terms of how the system obtains 3D coordinates controlled by humans: physical sensors, computer vision, pre-defined courses and a microphone array.

We list one example for each category below, and make a comparison after the list.

1. Physical sensors

   Figure 1.1 is showing a work from PerMMA [1]. The robotic arms are controlled via a mechanic device.

   The man in the wheel chair controls the robotic arms via a control stick holding in his right hand.

Figure 1.1: A wheel chair with robotic arms

In submarines, spacecraft and bomb disarm robot, this kind of technique is used.

This kind of sensor uses similar technology to that found in old fashioned mechanical mice.

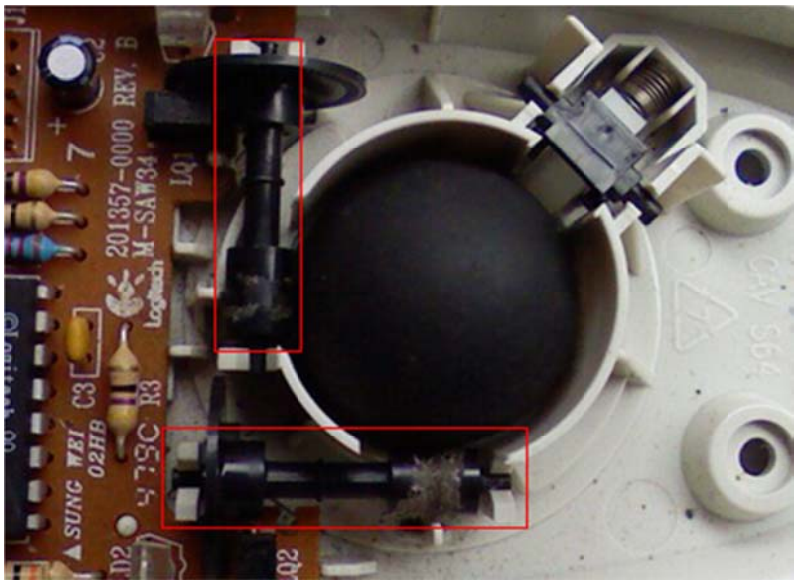

Figure 1.2: An internal view of a mechanic mouse

The cost of building this kind of device can be very high, when accuracy requirements are high. However, the cost of the localization part is limited.

2. Computer Vision

   Figure 1.3 shows a man from Tsukuba University demonstrating the "Gesture Interface" [2] system.

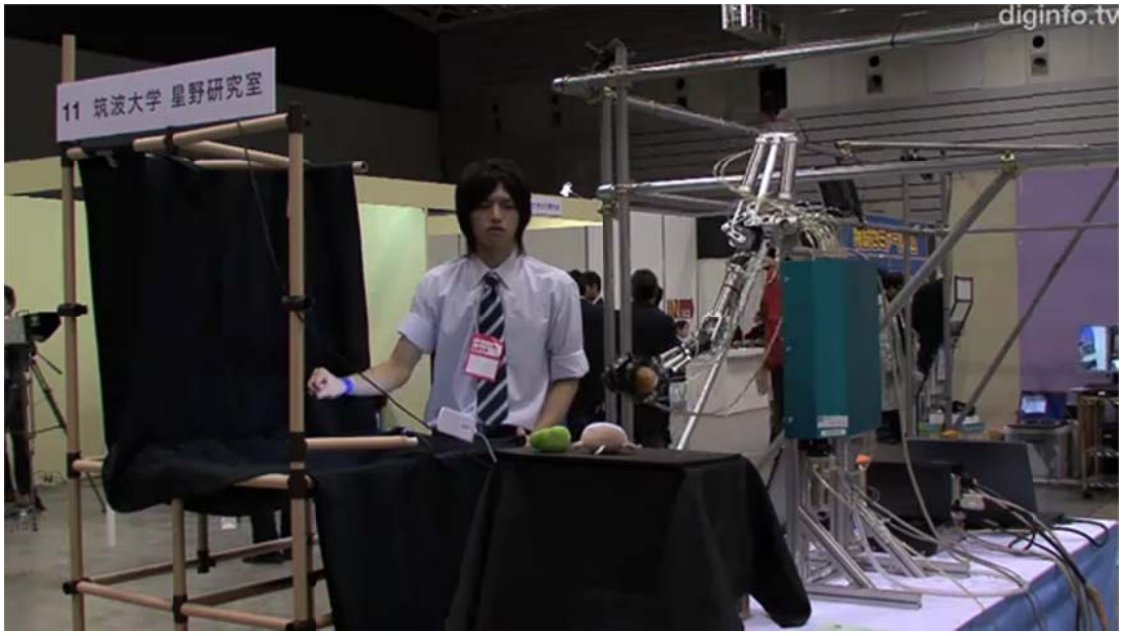

Figure 1.3: A camera based localization system

This device from Hoshino Lab in Tsukuba University demonstrates amazing 3D position control and hand gesture recognition using only two cameras.

This technique is still under research. From the original video clip, we can see there is obvious delay from hand movement to the robotic arm movement, although it claims the computer updates the data every 10 milliseconds. We have to assume that the delay is on the mechanical side, rather than the localization side. Thus this is a real-time system.

Another localization-like system using a simple vision technique is the optical mice. It can be very accurate and responsive. Since the outputs are relative coordinates, which cannot be treated as a real localization system.

3. Pre-defined Course

   Figure 1.4 is a photo taken in the Martin Memorial Hospital South [3]. The doctor is using a surgical device known as "MAKOplasty" [4] to perform knee surgery. It allows the doctor to pre define which parts of the knee need to be replaced, and operates on those parts only with very high accuracy. Without this device, the whole knee must be replaced in the surgery.

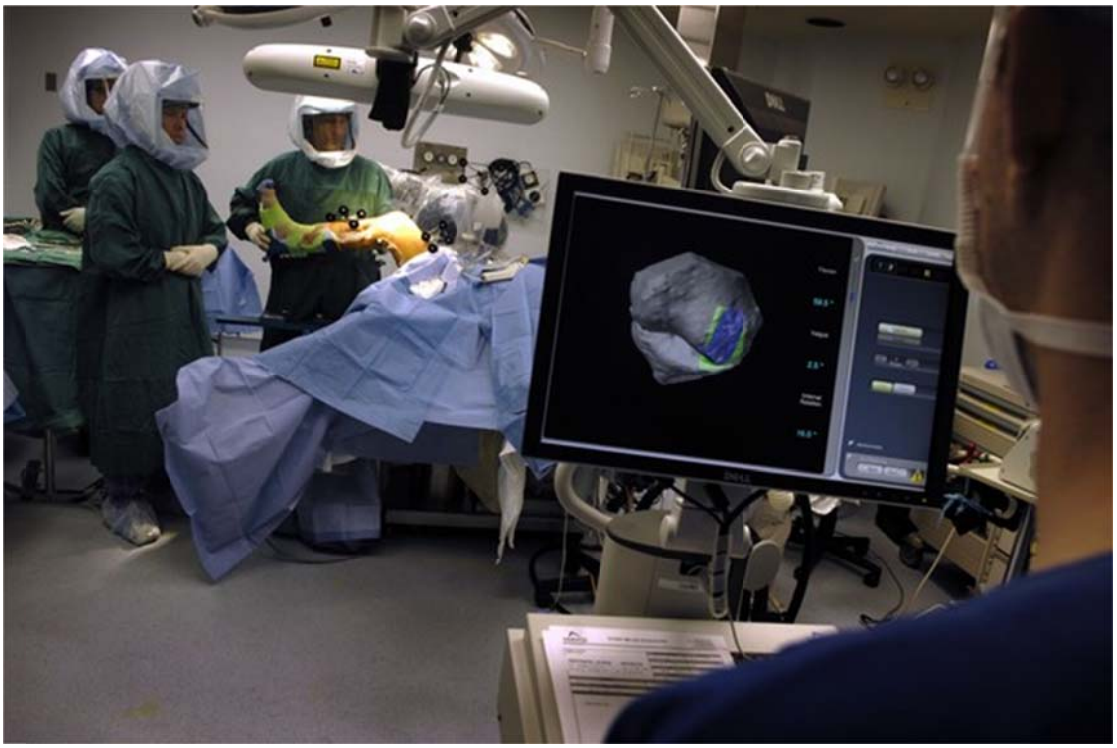

Figure 1.4: A robotic knife operating on a predefined course

This method can achieve extremely high accuracy. The difficult parts are the 3D image pre-acquisition and precise mechanic controls.

A high definition 3D image must be captured in advance as a reference for doctors to determine the course of the operation.

The nature of this technology determines that it cannot be operating in real-time. So its application is almost certainly limited to medical surgery.

4. Microphone Array Systems

This kind of system uses microphones as sensors, to locate the sources which generate sound.

Figure 1.5: A voice tracker based on a microphone array system

Figure 1.5 is an example of a voice tracker system from Acoustic Magic [5]. It is used in videoconferencing, to locate the speakers. However, this kind of system usually works only when the speakers are relatively static (sitting around a table).

Another type of systems using microphone arrays can deal with real-time moving source localization, which is the area we are exploring in this thesis. An overview of such systems can be found in the Related Work section of this chapter.

For a summary of the above technologies, please refer to **Table 1.1**

	Precision	**Cost (US$)**	**Real-time**	**Multiple Source**
Physical sensors	Widely ranged	Several to hundreds	Yes	Mostly No
Computer Graphics	Ranged from micrometers to cm	Less than 100	Yes	Yes
Pre-define Course	In micrometers	Thousands	No	N/A
Microphone Array Systems	In hundreds micrometers to several mm	Several	Yes	Yes

Table 1.1: Comparison among localization technologies

Our objective is to investigate how accurate we can achieve under "real-time" conditions, using devices such as inexpensive microphones. By "accurate" we mean the probability of correctly measuring a position of a moving point source in 3D space must be high. By "real-time" we mean the system should finish the measurement within 20 milliseconds. By "inexpensive" we mean that aside from the computational unit, which can be a computer or a microchip, the whole system costs less than 15 USD.

1.2. THESIS ORGANIZATION

Chapter Chapter 2 gives a general review of previous theoretical and experimental work related to sound source localization. Chapter 3 reviews models and algorithms studied and used in the time delay estimation (TDE) area, to found a base for our research. Chapter 4 discusses the resolution problem in TDE and its importance, which is also our focus. Chapter 5 describes the four tests used to solve the aforementioned problem. Chapter 6 is a relatively independent chapter describing a simulation tool implemented to prototype the energy

propagation method other than TDE. Chapter 7 and Chapter 8 state the implementation of the application and our experimental results. We draw conclusions on the whole research in Chapter 9.

CHAPTER 2

RELATED WORK

There are mainly two robust approaches in performing sound localization:

1. Steered Beamforming (SB) [6]
2. Time Delay Estimation (TDE) / Time Difference of Arrival (TDOA)

In the first approach, the orientation of the sensor (microphone) array may be changed as a part of the localization process. Since we have chosen fixed sensor positions, it is not applicable. The advantages of fixed sensor positions include low hardware complexity and low cost, which imply wider applications.

We discuss various TDE based works below. More details are discussed in the TDE Methods of Chapter Chapter 3.

Kleeman and Kuc (1994) [7] presented a two dimensional localization and classification system in indoor conditions with an ultrasonic sonar array for mobile robots. It can accurately classify and localize planes, corners and edges in an eight meters range.

Brandstein and Silverman (1996) [8] confirmed the practicality of using TDE methods to perform sound source localization with limited computational power by experimenting in real room conditions.

Merging sound source localization with other techniques to enhance the result has been studied by Nakadai et.al (2001 [9], 2002 [10]). Trifa et.al (2007) [11] further studied the advantages and disadvantages of four sound source localization methods based on TDE. They

studied generalized cross-correlation (GCC) [12], GCC with Phase Transformation (PHAT), Moddemeijer Information theoretic delay criterion (MODD) [13] and Cochlear filtering (COCH) [14]. PHAT had the greatest accuracy, but not fit the model as well as COCH, which performed adequately under reasonable SNRs. The conclusion was that COCH was the primarily considered method.

Valin et.al (2003) [15] developed a three dimensional localization system with eight microphones based on TDE, which can achieve 3° precision in a three meters range.

Han Yi and Wu Chu-na (2010) [16] demonstrated another work using three high precision sensors as shown in **Figure 2.1**, to perform real-time sound localization for a moving sound source, under the condition that the sound source be controllable. The precision is ±1cm.

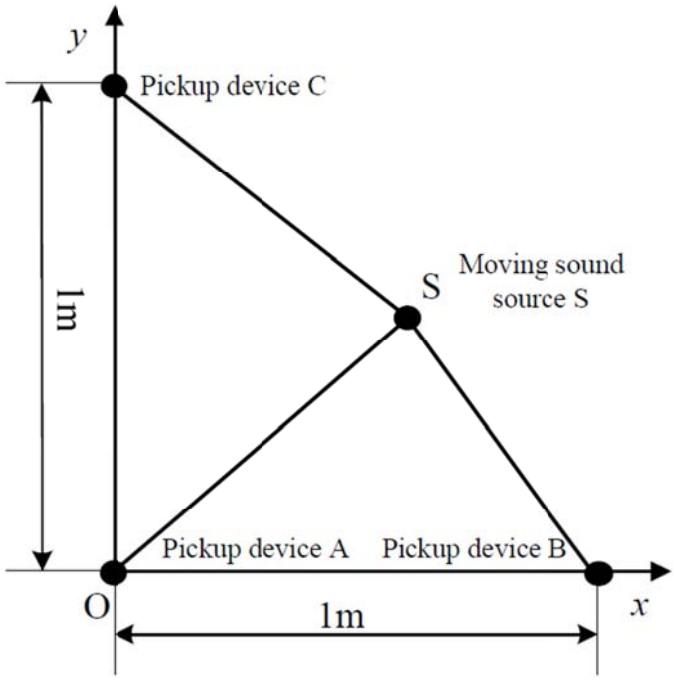

Figure 2.1: The devices configuration

In their work, the moving sound source emits a sound signal for 0.3 seconds and stops for two seconds, in order to let the environment return to a quiet state.

Duo pen [17] is a product which measures 2D position changes in real-time. It is used as an accessory to enable the handwriting ability of non-touch-screens and normal paper smaller than A3 size (inclusive). It has a pen-like device which emits two types of signals: infrared and ultrasonic wave; and another component of one infrared sensor and two ultrasonic sensors to detect the time difference of the arrival of the two kinds of signals, and calculates the location.

After time delay is estimated, how to do the localization is another stream of research. Some of the work [7, 17] can be computed by solving a linear system; others must be solving a hyperbolic localization problem. Chan and Ho (1994) [18] presented a method that makes the problem possible to be solved efficiently with computers.

CHAPTER 3

TDE METHODS OF SOUND SOURCE LOCALIZATION

In this chapter we first analyze why we have to use TDE (Time Delay Estimation) methods, and then review the TDE methods which have been studied in the past. In the review, we divide the content into two sub-sections: first we discuss some mathematical models for simulating the sound transmission; then we briefly introduce algorithms studied in the past to do localization based on certain models.

3.1. WHY TDE?

Sound signals contain multiple kinds of information. Here we consider amplitude and phase for our localization method.

From our experiments conducted in Chapter Chapter 8, there are many factors which are introduced by the status of the source. For example, the orientation of the sound source if it is not a point source or if we are holding the sound source, the blocking effects of the hands. As a result, methods based on amplitude turn out to be unstable, or not very robust.

Methods based on phase information are time delay estimation methods. This kind of methods measures the difference in time between the emission and arrival time of sound signals, or between arrival times of sound signals between different sensors. The first is TDE (time delay estimation), and the second is TDOA (time difference of arrival). They appear differently, but they use the same techniques. For simplicity and unless their differences absolutely matter, we do not distinguish them from each.

TDE has been extensively studied for over thirty years. In the next sub-section, let us review some of the models and algorithms studied which are relevant to this thesis.

3.2. MATHEMATICAL MODELS OF SOUND PROPAGATION

There are in general three models [19] to describe the acoustic environment in TDE. They are the ideal single-path propagation model, the multipath model and the reverberation model.

3.2.1. IDEAL SINGLE-PATH PROPAGATION MODEL

This model assumes that the signal is only affected by attenuation, as well as the phase shift caused by the distance between source and sensor. We formulate this using equation **(1)**:

$$\mathrm{x[k]} = \mathrm{a} \cdot s[k - t] + w[k] \qquad \textbf{(1)}$$

In equation **(1)**, we assume that we have only one sensor x. x[k] is the signal received by x at time instant k; a is the attenuation factor, which is a scalar, often given by $\frac{1}{d^2}$, where d is the distance between source and sensor; s[k] is the source signal; t is time delay between source and sensor, which models the phase delay; and w[k] is the noise signal received. The noise signal is often modeled as a zero-mean random signal independent to the source and the sensor.

3.2.2. MULTIPATH MODEL

The single-path model only considers the direct-path of the signal. However, in reality, due to reflections of sound, each sensor receives the same signal multiple times, but with different attenuation and delay. Thus we need this multipath model to take that fact into consideration.

We can formulate this model with equation **(2)**:

$$x[k] = \sum_{n=1}^{N} a_n \cdot s[k - t_n] + w[k] \qquad \textbf{(2)}$$

In equation **(2)**, N is the number of rebound replicas of the sound signal we want to model, a_n is the attenuation factor just for the n-th replica; t_n is the time delay for the n-th replica.

Figure 3.1 is a simple illustration of this model:

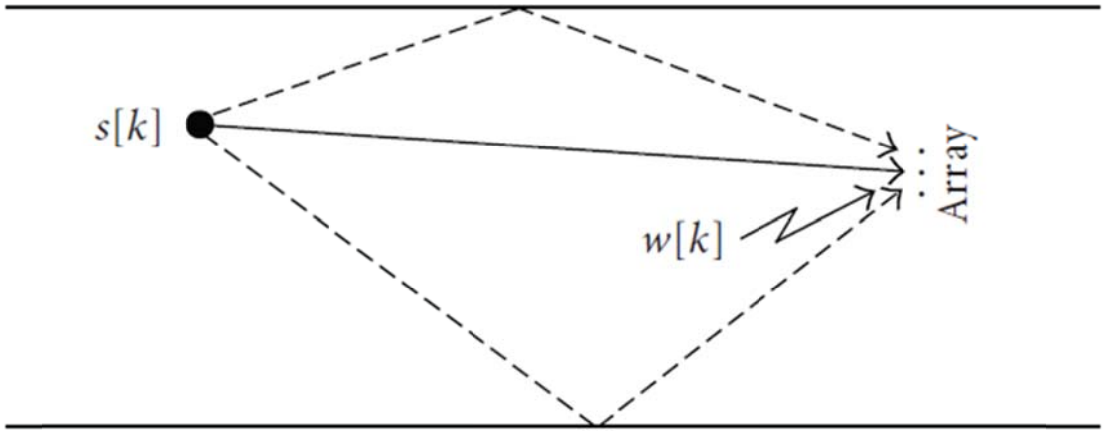

Figure 3.1: The source signal is reflected from two boundaries that produce two replicas of the original signal

3.2.3. REVERBERATION MODEL

The multipath model does not work well when the number of paths is large, because it is difficult to estimate all t_n in equation **(2)**. In indoor conditions, the walls and ceilings produce echoes which will confuse the sensor [20, 21, 22]. Hence a more realistic model: the reverberation model [23] was proposed without the need of pre-estimating any t_n.

Since there is no time delay explicitly expressed in this model, there will not be any plain solutions. We have to locate the signal impulse for each path, and obtain the TDOA information by identifying two direct paths between any two sensors. The computational cost is potentially large because of the first step.

3.3. TDE ALGORITHMS

Various TDE algorithms have been studied in the past. We cite only some of the critical ones.

3.3.1. CROSS-CORRELATION METHOD

The cross-correlation (CC) method is the earliest and most straightforward TDE algorithm. Without loss of generality, we assume we are only examining two sensors x_0 and x_1 for simplicity. We further assume that the signal received by x_n is $x_n[k]$.

This algorithm computes all cross correlation values for each reasonable delay Δ, and outputs the delay which results in maximum cross correlation. We formulate the problem in equation **(3)**:

$$\hat{t} = \arg\max_{\Delta} E\{x_0[k]x_1[k-\Delta]\} \quad \textbf{(3)}$$

$\hat{t}$ is the estimation of the true delay t in equation **(1)**. E{*} stands for mathematical expectation. Usually we assume an average distribution for E, thus the equation becomes:

$$\hat{t} = \arg\max_{\Delta} \frac{x_0[k]x_1[k-\Delta]}{\|\Delta_{max}\| + \|\Delta_{min}\|} \quad \textbf{(4)}$$

We replace the mathematical expectation with an averaging function.

Other versions of this method are formulated from the average-magnitude-difference function (AMDF) and average-squared-difference function (ASDF) [24]. We omit the details here because the concepts are very similar, just by replacing * in E{*}.

There is also a generalized cross-correlation (GCC) method [12] made popular in 1976. It operates on frequency domain and has more parameters to be tuned, making it more flexible.

3.3.2. LMS-TYPE ADAPTIVE TDE ALGORITHM

The difference between this method [25] and CC methods is that this method measures the time delay by minimizing mean-square-error (MSE) of $\mathbf{x}_0$[k] and an FIR filtered version of $\mathbf{x}_1$[k]. The time delay is given by the lag time of the largest component of the filter.

To be clear, we formulate this algorithm.

Assume we have two signal vectors received at two sensors $\mathbf{x}_0$[k] and $\mathbf{x}_1$[k], the length of each signal vector is L. An FIR filter of length L is **h**[k]. Equation **(5)** gives the error function:

$$\mathbf{e}[\mathrm{k}] = \boldsymbol{x}_0[k] - \boldsymbol{h}^T[k]\boldsymbol{x}_1[k] \qquad \textbf{(5)}$$

In order to estimate **h**[k], we need to minimize $\mathrm{E}\{\boldsymbol{e}^2[k]\}$. If we use an adaptive algorithm, **h**[k] can be estimated by

$$\mathbf{h}[\mathrm{k}+1] = \mathbf{h}[\mathrm{k}] + \mu \mathrm{e}[\mathrm{k}]\boldsymbol{x}_1[k] \qquad \textbf{(6)}$$

μ in equation **(6)** is a customizable small adaption step size.

Given this estimation, the largest component h[s] in **h**[k] is the component we are looking for, and its order "s" is the time delay estimation result.

3.3.3. FUSION ALGORITHM BASED ON MULTIPLE SENSOR PAIRS

The idea of fusion algorithm is using redundant information to generate more accurate results [26]. For example, in 1D localization problem, we use three linearly aligned sensors instead of using only two; then we can have three pairs. Obviously the three pairs are not independent, because $\Delta_{0,1} + \Delta_{1,2} = \Delta_{0,2}$.

The performance gain of this algorithm mainly comes from better estimation with noise and reverberation.

This algorithm has in general two stages. The preprocessing stage measures the time delay independently by GCC methods. The post-processing stage uses transformation functions to consolidate the redundant information.

3.4. SUMMARY

We have discussed single-path, multipath and reverberation models which are used to describe the acoustic environment mathematically. We also have mentioned three kinds of algorithms to solve the TDE problem.

Apparently, there are not only three kinds of algorithms. Those algorithms operate under multi-channel situations (more than two sensors) are not described in detail, such as multichannel cross-correlation algorithm [27], adaptive eigenvalue decomposition algorithm

[23] and adaptive multichannel time delay estimation [28]. Those methods are not widely used, but have potentials.

CHAPTER 4

INTRODUCTION TO THE RESOLUTION PROBLEM

In this chapter, we first describe the nature of the resolution problem; then we provide the algorithms we have tested to solve this problem, also comparisons between existing algorithms.

4.1. NATURE OF THE RESOLUTION PROBLEM

When dealing with digital signals, the data available is a collection of discrete samples. Therefore, the time delay measured by the algorithms described will be an integral multiple of the sampling period. For example, if the sampling rate is 44,100 Hz, which is the CD standard, the sampling period is 1/44,100 second. This means the accuracy is limited to 1/44,100.

In many situations, this accuracy is not enough. How to achieve finer resolution under the restriction of sampling rate has become a challenging problem.

4.2. POSSIBLE SOLUTIONS TO THE RESOLUTION PROBLEM

There are in general three types of solutions: using a higher sampling rate, increasing the distance of sensors, and interpolation. We look at these solutions one by one.

4.2.1. USING A HIGHER SAMPLING RATE

The most direct solution is using a higher sampling rate. However this is not always an option with the hardware limitations of both the sensors and the computational power.

Although a higher sampling rate means a shorter sampling period, more samples within a certain interval and finer resolution, it also means more data to process, a higher standard of building sensor and higher power requirements.

4.2.2. INCREASING DISTANCE OF SENSORS

Here we use an example to illustrate why this method works in the context of direction of arrival (DOA) measurements.

In **Figure 4.1**, sensor 0 and sensor 1 have distance *d*. A sound source s[k] is placed far afield from the sensors. The purpose of DOA is measuring the angle θ by measuring the time delay of sensor 1 from sensor 0. Also, the angular resolution is defined by how many DOA measurements can be made between 0 and π. The angular resolution determines the ability of the system to separate two closely placed sound sources.

We have already assumed that the distance between sensor 0 and sensor 1 is *d*. We further assume the wave propagation velocity is *v* and the sampling rate is *f*. It is obvious that the maximum and minimum time delays Δ that can be estimated are *df*/*v* and –*df*/*v*. The bearing angle θ is given by $\theta = \arccos\frac{v*\Delta}{d}$.

Therefore, the resolution of measuring θ depends on how many different values can be measured for Δ (the time difference of arrival at the two sensors) in [-*df*/*v, df*/*v*]. Increasing the distance of the sensors means increasing d, which will lead to a wider range for Δ. If we keep the sampling rate constant, there will be more samples taken in the range of Δ, which means more measurements of θ.

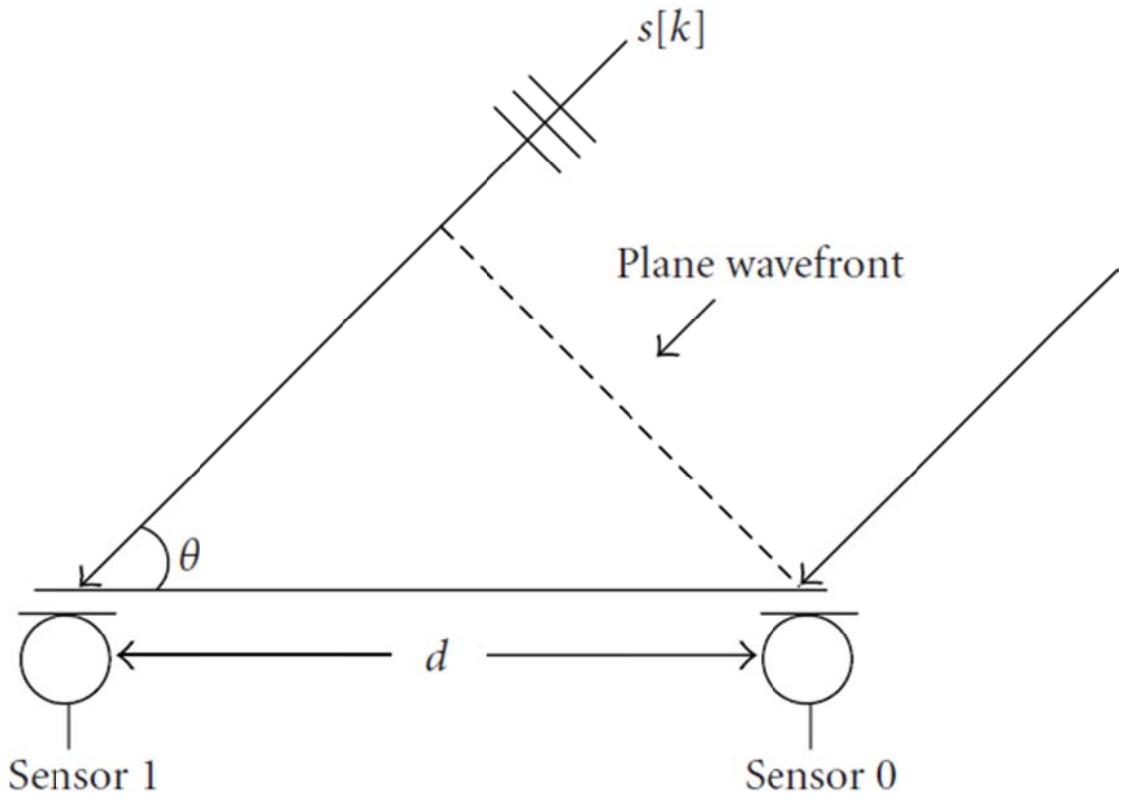

Figure 4.1: 2 sensors with a distant sound source

However, the implementation of this method is constrained by the space limit of the application.

4.2.3. INTERPOLATION

Interpolation methods are flexible; and in most cases, it is not necessary to build finer hardware.

As a matter of fact, interpolation also requires additional computational power. However, the additional computational power needed is often much less than the improvements they can achieve, depending on the algorithm. Also, many algorithms are designed to be tunable, one can choose between speed and resolution by tuning the parameters. For example, increasing

the decrease the window size in GCC based methods can result in faster computation but lower accuracy.

The solutions based on interpolation can be fitted into different categories.

In 1993, Giovanni proposed a method named "parabolic interpretation" which does not require prior knowledge of the sound source [24**Error! Bookmark not defined.**]. His method can also be viewed as a resampling process as described in the IEEE Signal Processing Magazine in 1996 [29].

Let us illustrate this method with an example.

Assume we have two signals in discrete form at sampling rate *f*. Therefore, the possible outputs of conventional methods are $\frac{n}{f}, n \in Z$. Sometime we need to measure the time delay more accurately when the hardware is limited. However, there is no sample taken at (n+ξ)/*f*, where $\xi \in (-1,1)$. So we need to do interpolation based on the sample near (n+ξ)/*f*, to achieve finer resolution.

Giovanni's method interpolates by approximating the outcome of the cross correlation with a continuous parabolic function. Once the integral offset with maximum correlation found, the task remains for the algorithm is to use the approximated parabolic function to estimate the location of the maxima.

Interpolation methods are very sensitive to noise, as the SNR increases, the variance in the result decreases exponentially. In our experiment, the algorithm will return perfectly precise

results when a digitally generated signal is used as the input (the signal does not come from any recording devices).

4.3. THE RESOLUTION PROBLEM WHEN THE SOUND SOURCE IS CONTROLLABLE

The interpolation method mentioned above does not require the source signal to be controlled by the system. In fact, whether the source signal is controlled by the system depends on the application. For example, a "sound pen" is a coordinates input device which only measures the location of the pen. The manufacturers can control everything in their system. Therefore the signal emitted by the pen can also be controlled and used as an advantage.

By knowing the exact source signal in advance, which is different from Giovanni's method, higher accuracy (lower variance) can be achieved.

4.4. SUMMARY

This chapter introduces the resolution problem caused by the high requirements of accuracy and the relatively low precision of hardware.

There are three major approaches: increasing sampling rate; increasing distance between sensors; and interpolation. Please refer to **Table 4.1** for comparisons.

	Restricted by Space	**Need Accurate Hardware**	**Need Additional Computational Power**	**Require Prior Knowledge of the Source**
Increase Sampling Rate	No	Yes	Yes	No
Increase Distance between Sensors	Yes	No	Yes	No
Interpolation	No	No	Yes but tunable	Yes

Table 4.1: Comparison among methods of solving the resolution problem

CHAPTER 5

SOLVING THE RESOLUTION PROBLEM WHEN SOURCE IS CONTROLLABLE

In this chapter, we formulate the resolution problem when the source is controllable, and then we discuss the difficulties as well as the possible solutions in detail.

5.1. PROBLEM SETUP

The reason we are studying this problem is because there are types of applications facing the similar situation.

The scenario we setup for this problem is a "sound pen". It is an absolute 2D or 3D coordinates input device, like a touchpad. The system contains of two parts: the "pen" and the sensors. We assume the "pen" is controlled by the software to make any kinds of sound when necessary; and the sensors are a set of microphones, which we can easily buy from electronics stores. The analog to digital converter (ADC) device for the microphones is as cheap as 0.8 US dollars per microphone. Building the whole system should cost less than 20 US dollars (8-12 sensors) for 3D inputs, and less than 15 US dollars (6-8 sensors) for 2D inputs.

5.1.1. ACCURACY REQUIREMENTS

Under a sampling rate of 44100Hz, sound can travel for 0.014 cm (assuming the speed of sound is 330m/s in air). The accuracy required for TDE in this application is that a measure will be different from the true value for less than 0.014cm, or 0.019 samples, at probability of 85%.

5.1.2. REASONS FOR THE REQUIREMENTS

We assume the "pen" should be used for 2D screen drawing, and the sensors are placed at four corners and the middle of the two long edges. We further assume that the screen will have a resolution of 1280 by 720 pixels, and 17 inches in size. We can calculate that the minimum resolution required for the system is roughly 0.014 cm, when the "pen" is near middle of the short edges or the center of the screen and moving towards the center. Converting 0.014 cm into a number of samples, we can yield the result, which is 0.019 samples.

5.2. POTENTIALS

The major difference between this problem and problems which assume the sound source is unknown is that we are potentially able to use the prior knowledge of the sound source to improve the results.

Like the work done by Douglas in 2002, which utilize a quadrature signal as the reference signal, to enhance the results [30].

5.3. CHALLENGES

The difficulties come from the inexpensive device we are using. For such devices, the reception SNR is low, and the frequency response patterns are very different from sensor to sensor. Moreover, the sampling rate is relatively low comparing to the resolution required.

Also, the quality of sound source is poor. This introduces another kind of instability to the system.

5.4. OUR TESTS

There are multiple tests we have made to solve this problem. We describe those tests in this section, and provide implementation and results in the Chapter Chapter 7 and Chapter Chapter 8 respectively.

5.4.1. ALGORITHM OVERVIEW

Our algorithms are all based on the cross correlation method described in Chapter Chapter 3.

Since the sound source is controllable, which means the received signal is a scaled version of the original signal; we know the exact frequency and type of the source signal.

Making use of the facts above, we are able to design band-pass filters to eliminate most of the noise in other frequencies. Moreover, different from the other problems, we do not need to use interpretation. Instead, we can use the generated continuous signal as a reference, to compute the correlation with each received signal. Finally, we do a subtraction of the computed values.

The sound source selected is a fixed frequency periodical sine wave, released by a cell phone manufactured in 2003 to simulate the electronic buzzer with poor quality.

5.4.2. GENERAL STEPS

The goal of these algorithms is to measure the relative time delay of the received signal to a reference time stamp. For example, in our experiments, we are measuring the time from when the first zero point at the beginning of a full sine wave appears, relative to time zero. Assume the signal is continuous, a zero point means the point which has zero amplitude.

Figure 5.1 illustrates this measurement. Note that the zero point is usually not sampled exactly.

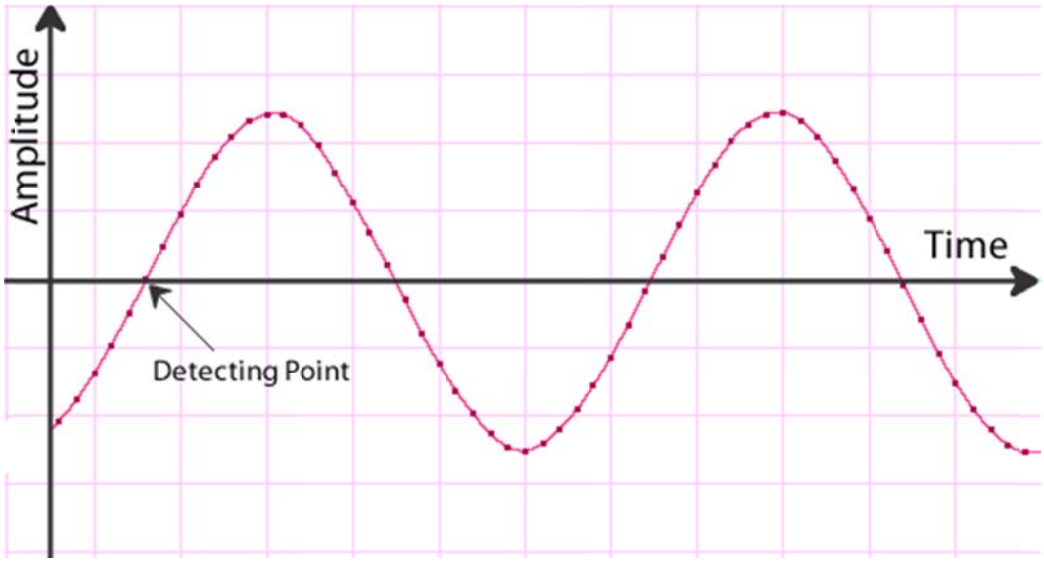

Figure 5.1: Illustration of time delay measurement. The measured time delay is the number of samples between the zero time and the detecting point (can be fractional number)

There are three major steps in these algorithms after the filtering process. The general idea is that we first locate the samples which appear to be near the first zero point, then use another step to refine this result.

STEP 1: ROUGH LOCATING

The input is an array of float numbers. We can use equation **(7)** to locate the zero points:

$$\forall\, i = 1..\mathrm{N}-1, report\ \{S_i, S_{i+1}\} s.t. S_i * S_{i+1} < 0\ AND\ S_i < 0 \qquad \textbf{(7)}$$

The algorithm in this step multiplies two adjacent samples and reports those pairs when: 1) the product is less than 0, and 2) the first element is less than 0. This will report all samples near the zero points which are at the beginning of full sine waves in time ascending order. We name the reported samples as "refinement boundaries", because they are inputs for the refinement step.

STEP 2: SCALE FACTOR ESTIMATION

In order to do cross correlation, we must have the source signal represented in the form of equation **(8)**:

$$\mathrm{s[k]} = \mathrm{Asin}(\mathrm{T} \cdot \mathrm{k} + \varphi) \qquad \textbf{(8)}$$

We already know the value of T, because the signal is generated by our system; and we are determining ϕ. The only unknown we need to know before determining ϕ is the scale factor A.

Here we must introduce one assumption: the sound source does not move significantly in 0.01 seconds. The reason for making this assumption is that we segment the sample array into length of 512 elements, and do measurements for each segment. The duration for 512 samples is roughly 0.011s depending on the sampling rate. The assumption is fair because the application requires the pen to be controlled by human hands. Since human hands will not move too fast, except in some extreme cases.

Under this assumption, we can estimate the scale factor by finding the sample which has the maximum absolute value.

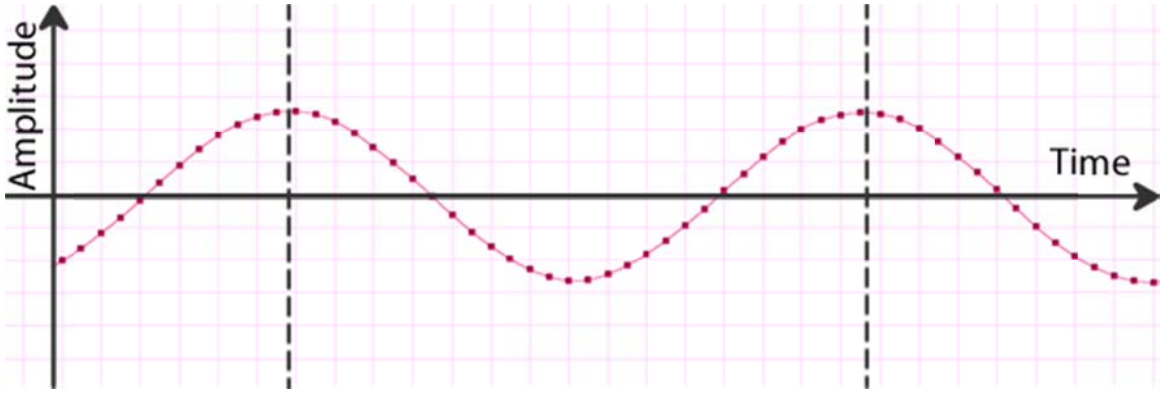

Figure 5.2: Illustration of estimating the scale factor. Dashed lines are located at the ground truth maximum values. The closer the sample to the black lines, the larger the value it has.

STEP 3: CORRELATIVE REFINING

Here we completed the preparation step. The cross correlation algorithm can be applied within the refinement boundaries.

5.5. TESTS FOR OPTIMIZING RESULTS

There are infinite possible ways to organize the algorithm because there are 512 samples we can make use of. The General Steps section gives the general framework of the algorithm. The following discussion is based on that framework.

We carry out experiments to evaluate the results of each test. Those experiments are described in a separated chapter (Chapter Chapter 8). In order to be brief and clear in the following description, let us define terms and symbols in **Table 5.1**.

Term/Symbol	Description
T	A full period of the periodic wave in radians. For example, the T for a sine wave is 2π
Wavelet	A segment of the periodic wave which has exactly the length of one wavelength, starts at $t = N \cdot T$
Channel/Input channel	Our signal is acquired by several microphones, each microphone is treated as one "channel", or "input channel"
Cycle	We mention that we segment the sample array into length of 512 elements In the description of Step2 in Section 5.4.2. Here we define each segment as a cycle. It is a closer view from implementation, because the hardware updates its buffer in cycles, and each cycle we are returned with 512 samples
Conventional	This word means the name of method mentioned afterwards will be limited to those methods listed in TDE Methods of Sound Localization chapter. The basic characteristics of those methods is that their outputs are integral numbers

Table 5.1: List of Symbols Used in Chapter 5

There are several assumptions we must make before going to the descriptions. All assumptions apply to all tests unless an exception is explicitly mentioned:

1. Tests only deal with one cycle, and each cycle is independent from other cycles.
2. Tests are limited to determine the time delay of only one channel from the start of a cycle.
3. Tests are elaborations of **Step 3**.
4. The step size of correlation refining is set to be 0.02 samples.

5.5.1. TEST 1: DIRECT CROSS CORRELATION AND MEAN METHOD

In this test, we apply CC for each wavelet, use equation **(3)** to calculate the correlation. Then we take an average over all result of all wavelets.

This method treats each wavelet as a single measure, and averages among all measures. It is a good method to yield finer precision when the step size is limited by the computational power.

5.5.2. TEST 2: DIRECT CROSS CORRELATION FOR-ALL METHOD

This test is based on the conventional CC method, but has a fractional step size. Since each sample has 16 bits precision, the fractional step size CC can give good results in theory.

This time we still use equation **(3)**.

5.5.3. TEST 3: SUM OF DIFFERENCE AND MEAN METHOD

The propagation model we are using is the ideal single-path propagation model defined in equation **(1)**. The noise term w[k] is assumed to be a zero-sum random number independent from the signal. In this test we use this assumption. Our goal is to seek a time difference, which results in smallest sum of difference.

The algorithm operates in exactly the same way as **Test 1**, but we replace equation **(2)** by equation **(9)**. In other words, we are trying to maximize the value of equation **(9)** other than equation **(3)**.

$$\hat{\mathrm{t}} = \arg\max_{\Delta} E\{-(x_0[k] - x_1[k-\Delta])\} \qquad \textbf{(9)}$$

Equation **(9)** uses the "zero mean" assumption, that when the sum of difference is close to zero, 2 signals have the same phase, and the difference is caused only by noise. The difference from AMDF is that there is no absolute value taken.

Note that this method works only under the precondition that Step1 has been correctly performed. Otherwise the function may stop at another extrema when the phase difference is near T/2.

5.5.4. TEST 4: SUM OF DIFFERENCE FOR-ALL METHOD

It is a "for-all" version of **Test 3**, just like **Test 2**, which is a "for-all" version of **Test 1.** The only difference is that we are use equation **(9)** instead of equation **(3)**.

5.5.5. TEST 5 – TEST 8: ASDF / AMDF WITH MEAN / FOR-ALL METHOD

By replacing cross correlation estimator (equation **(3))** with average squared difference function (ASDF) and average magnitude function (AMDF), we can further derive four Tests, which are: Test 5 – ASDF Mean Method; Test 6 – ASDF For-All Method; Test 7 – AMDF Mean Method; Test 8 – AMDF For-All Method.

5.5.6. SMOOTHNESS FIX

This is not an independent test to the previous ones. The purpose is to stabilize the results by averaging the current output with the previous outputs. This simple step improves the standard variation by 50% from our experiments, and so matches the theory. However, this fix does not really improve the TDE measurement itself.

5.6. SUMMARY

In this chapter, we described a problem related to the resolution problem in TDE. The major difference is that the sound source is controllable, so we are able to know precisely the profile

of the signal, and even add extra information to help improve the results (although in our research, we have not found anything helpful to add).

We also developed a framework (general steps) for solving this problem. At last, we provided three tests to implement Step3 in the framework, and one fix to improve the overall smoothness of the output.

In the next chapter we describe another method we investigated for solving the sound source localization problem based on energy propagation.

CHAPTER 6

ENERGY PROPAGATION BASED METHOD

We have investigated another possible solution for solving the sound source localization problem at the very beginning of our work. In this independent chapter, we discuss this Energy Propagation Based Method, and finish the evaluation within this chapter.

We still use the same problem setup described as "sound pen" in Section 5.1.

6.1. OVERVIEW

The idea of this method is utilizing the amplitude ⇔ distance relation of sensors and the sound source, estimating the distance of the source to each sensor, establishing a linear system to find the location of the source. We describe the method in a systematic way in the following sections.

The devices we are using are described in Section 7.1, which are the same devices for investigating the resolution problem in TDE.

6.2. DISTANCE MEASUREMENT

We measure the distance based on the ideal path single propagation model. We further assume the sound source is a perfect point source, which has the amplitude ⇔ distance relation described in equation **(10)**:

$$A[k,d] \propto \frac{\rho}{d^2} \qquad \textbf{(10)}$$

In equation **(10)**, ρ is a constant related to the temperature, can be determined by calibration; $A[k,d]$ represents the amplitude received for source x[k] at distance d.

We calculate the total energy by adding the absolute amplitude of all samples received for a fixed period of time.

6.3. LOCALIZATION

After the distance to each sensor is measured, we are able to establish a linear system, with each sensor contributing one equation in the form of equations **(11)** in the 2D localization case:

$$\begin{cases} (x-x_0)^2+(y-y_0)^2={d_0}^2 \\ (x-x_1)^2+(y-y_1)^2={d_1}^2 \\ (x-x_2)^2+(y-y_2)^2={d_2}^2 \\ \cdots \end{cases} \quad \textbf{(11)}$$

In equations **(11)**, x and y are unknown 2D coordinates for the sound source; x_i and y_i are coordinates of the sensors, which are physically measured and inputted into the system.

The 3D localization case follows equations **(11)** naturally only by adding an additional component to the coordinate system.

In the 3D configuration described in Section 6.5, we use five sensors (see **Figure 6.3**). We further assume the sound source will always have positive Z coordinates. Thus only three equations are needed.

We divide the five sensors into four groups, by grouping the center sensor with each pair of adjacent corner sensors. We compute one set of coordinates for each group, and weight over the total energy received by each group. The reason is that the more energy received, the higher SNR is.

6.4. RETREIVE THE CLIPPED INFORMATION

When the sound source is close enough to the sensor, some of the samples exceed the limit of the sensible region, and are therefore "clipped" to the maximum/minimum value. **Figure 6.1** is an illustration of the occurrence of the clipping problem.

Figure 6.1: Illustration of the clipping problem. The clipped samples are the points that appear to be "attached to" the top and bottom. The grey lines are safe bounds for software playback, the top and bottom lines are hardware boundaries during the recording

There are two strategies to handle the clipping problem. One uses a lower volume sound source, the other is through estimation.

The first strategy is not applicable because we require the sound source to be loud enough so that the sensor at the far end can receive the signal dominated by the sound source, which will increase the accuracy and stability. Otherwise, noise will have a severe impact on the result.

We developed an estimation method to compensate for the clipping problem without constraining the hardware.

Observe the fact that the sine wave is always clipped symmetrically, let us only consider a half wavelength in $[0, \pi]$. If the samples in $[\varphi, \pi - \varphi]$ are clipped, we can still calculate the amplitude of this sine function based on the unclipped samples.

Recall that for a sine function

$$\int A \sin(x)\, dx = -A\cos(x) \tag{12}$$

We can compute the value of equation **(12)** from the samples:

$$\begin{aligned} A_\varphi = \int_0^\varphi A \sin(x)\, dx &= -\mathrm{A}(\cos(\varphi) - \cos(0)) \\ &= \mathrm{A}(1 - \cos(\varphi)) \end{aligned} \tag{13}$$

In equation **(12)** and equation **(13)**, A is the target we are estimating, and A_ϕ in equation **(13)** is the total amplitude which is not affected by clipping in $[0, \frac{\pi}{2}]$.

From equation **(13)**, we can represent A with A_ϕ simply by rewriting it to equation **(14)**:

$$A = \frac{A_\varphi}{(1 - cos(\varphi))} \tag{14}$$

In equation **(14)**, A_ϕ can be directly computed from the samples. Thus the value of A is also bounded.

Once we have the value of A, the estimated total amplitude can be evaluated by equation **(15)**

$$\begin{cases} A_{total} = \int_0^{2\pi} (A\sin(x)\,dx) * \text{AverageNumberWaves} \\ \text{AverageNumberWaves} = \dfrac{\text{BUFFERSIZE}}{SamplingRate * Frequency^{-1}} \end{cases} \quad \textbf{(15)}$$

A in equation **(15)** is the computed value from equation **(14)**. A_{total} is the estimated amplitude while clipping occurs. AverageNumberWaves is the average number of waves estimated within a BUFFERSIZE. BUFFERSIZE is the number of samples the hardware can capture within a certain period of time, which we set to be 0.011s (512 sample periods under sampling rate of 44,100 Hz).

6.5. EXPERIMENTS

The experiments are carried out in both 2D and 3D cases. **Figure 6.2** and **Figure 6.3** illustrate the setup of the sensors. The sensors are all located at the same plane, and we assume the sound source will not move below that plane. There was no particular preference for the orientations of the microphones, because our model does not consider the polar pattern of the response.

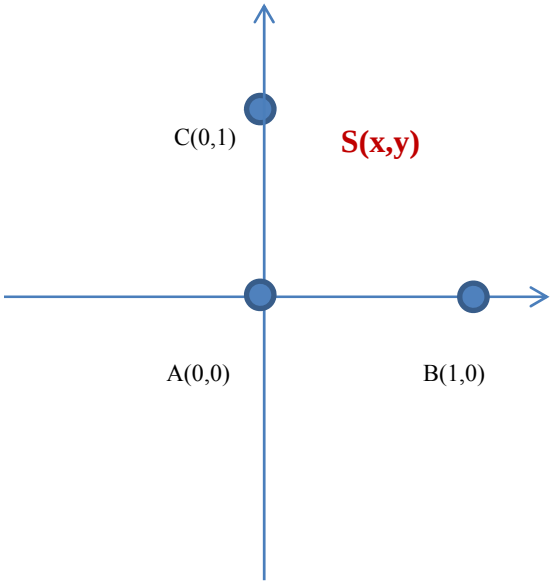

Figure 6.2: The setup of 3 Sensors for 2D Localization. S(x,y) represents the sound source

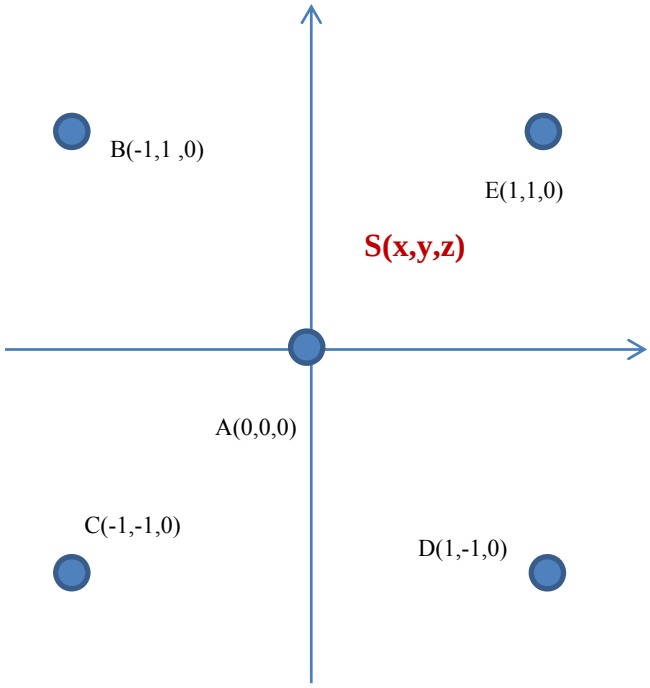

Figure 6.3: The setup of 5 Sensors for 3D Localization. S(x,y,z) represents the sound source in 3D space.

We are not able to achieve accurate results through this method, meaning that the output does not follow the actual movement of the sound source. In the 3D case, the output appears to be unpredictable. **Figure 7.2 - Figure 7.4** demonstrate one of the measurements.

6.6. CONCLUSION

We have considered two possibilities which affect the results significantly:

1. The microphones have directional polar patterns of response. In our experiment, the microphones have a cardioid polar pattern (see **Figure 6.4**). It is too complicated to model the pattern precisely and efficiently, and in fact, it requires accurate control of hardware, i.e., the precise angle of the microphones. So we had to choose a simple propagation model, otherwise the system would be too complex to fit our goal.
2. The hand movements and the shape of the sound source itself can both affect the propagation of the sound waves. In the same way, energy propagation suffers severely from the blocking effect caused by obstacles and the sound source itself.

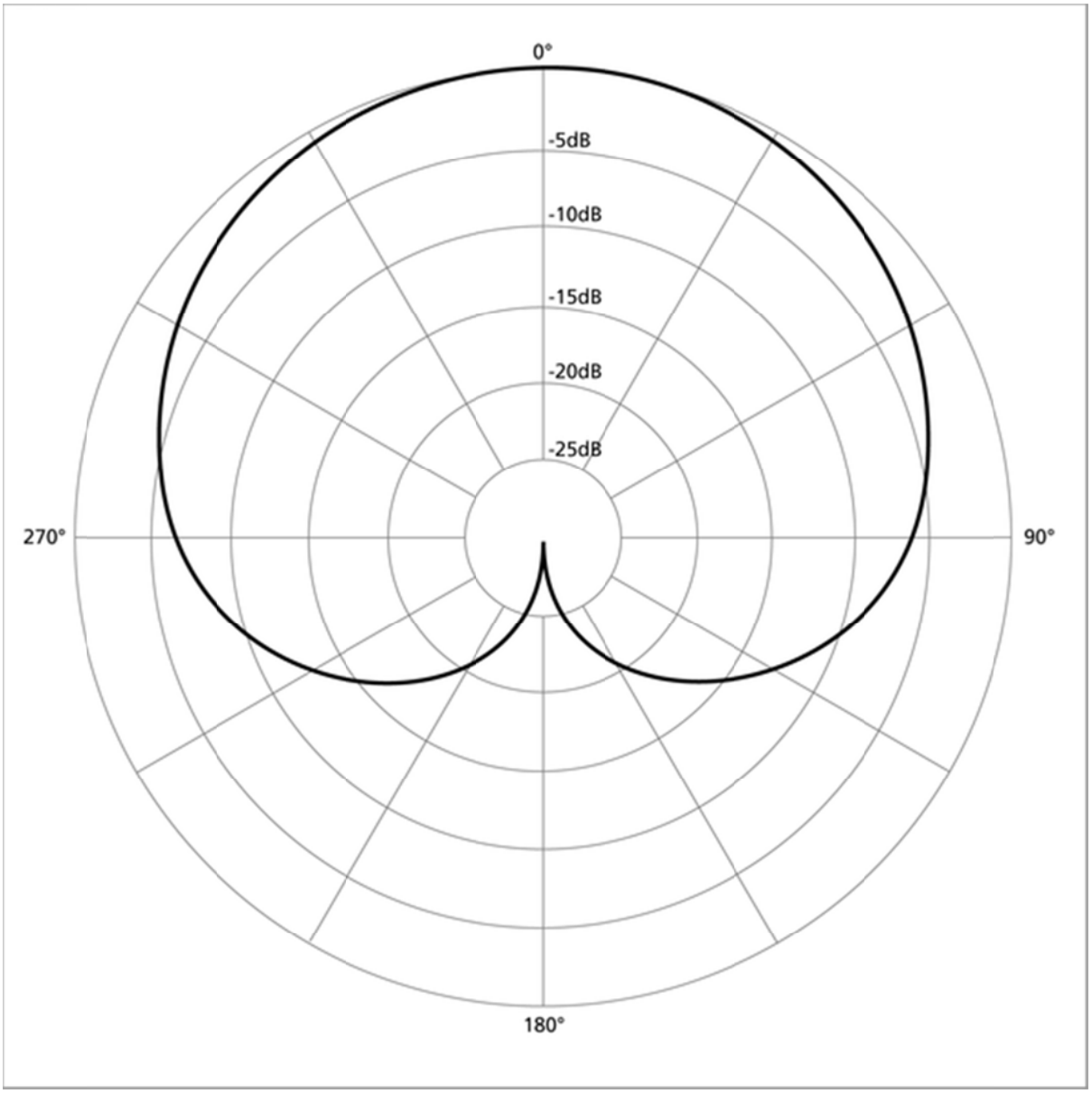

Figure 6.4: Cardioid polar pattern of microphone response. The diagram showsa top-view of a microphone. The polar coordinates indicate the relative response level of the microphone in different angles.

Unfortunately, there are no applicable solutions we can find in our research. For the first problem, unlike in the lab environment, it is not possible to have a perfect model for all kinds of situations in a real environment. For the second problem, no matter how we design the

hardware, the side effect caused by hands holding the sound source is neither preventable nor predictable.

CHAPTER 7

IMPLEMENTATION

There are two types of approaches we are investigating. They are TDE based methods and energy propagation based methods. The tool we built can test TDE methods in the offline mode, and prototype the 2D/3D localization system in an online mode for energy propagation based methods.

7.1. HARDWARE

- Nokia6600 which is used to emit a continuous sine wave of 1500Hz
- Five USB microphones with a fixed sampling rate 44100Hz
  - The microphones are in a cardioid polar pattern (see **Figure 6.4**)
  - Each microphone costs about two US Dollars, purchased from the Golden Computer Market [31] in Hong Kong
- PC:

  CPU: I7 930 @ 2.79 Hz

  RAM: 6G @ 1600 Hz

  Mother Board: Gigabyte X58 DS3

  HDD: Western Digital 1000FALS

  USB Port Version: 2.0

Please refer to **Figure 7.1** for a glance at the hardware:

(1) (2)

Figure 7.1: The mobile phone (1) and microphone (2) used in the implementation

Our test shows that the mobile phone can handle from 100Hz up to 3000Hz broadcasting much better than other frequencies. We selected the middle range at 1500Hz as the frequency for the experiments.

7.2. SOFTWARE

The software is built with Microsoft C# using .Net Framework 4.0. It uses NAudio [32] library to handle sound file I/O, and SlimDX Framework [33] to handle 3D rendering.

Figure 7.5 shows a flow chart of the program. Command Line Mode is an offline mode designed to test TDE methods. The GUI Mode is for the simulation of the amplitude based method.

Figure 7.2 is a screen dump of the 3D visualization window. The X, Y, Z axes are labeled by X, Y, and Z respectively. Four rods attached to X and Y axes, including the origin itself, are

positions of the microphones. The dot is the current position of the source. The "×" below the dot is the projection of the dot at XY plane; the "×" attached to Z axis indicates the current Z coordinates. The screen dump is showing a source location at [0.34, 0.78, 0.23], assuming the lengths of the three axis are 2 units.

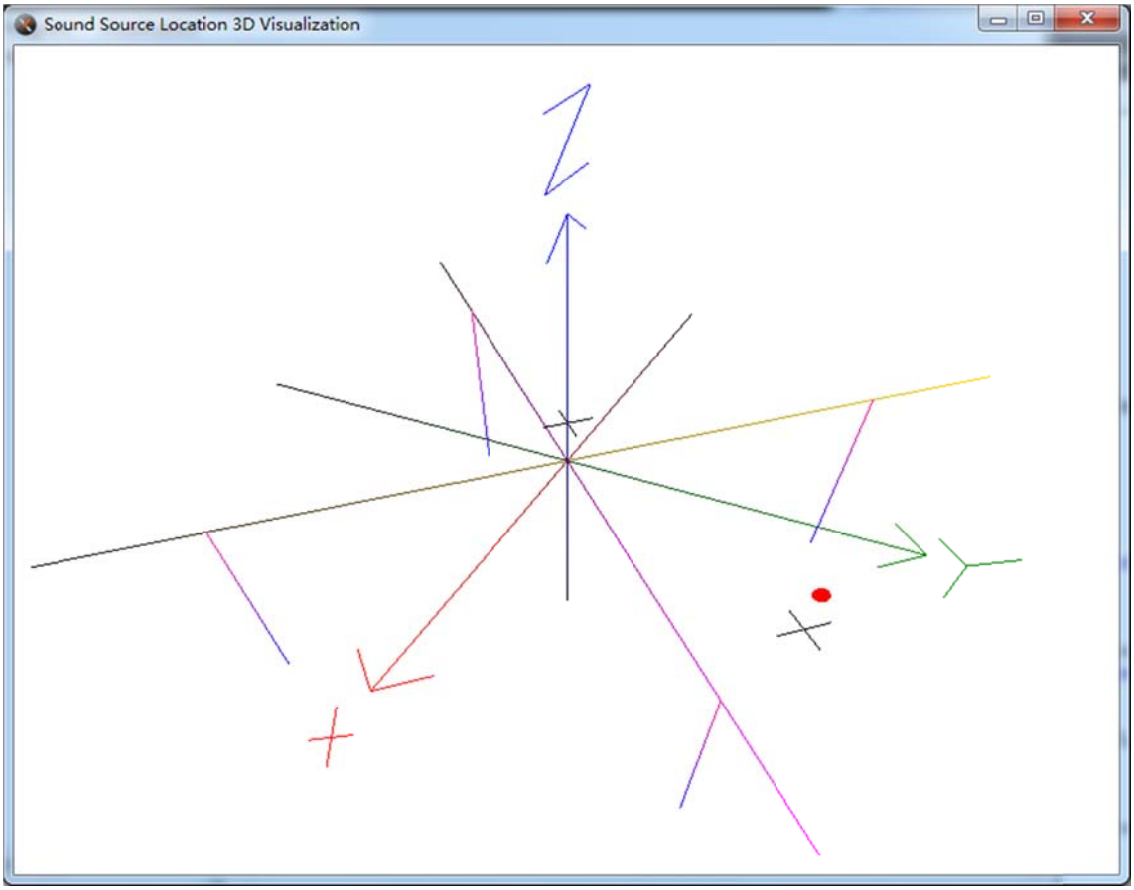

Figure 7.2: The detected position of the source [0.34, 0.78, 0.23]. XY plane is located right above the microphones, and parallel to the ground. Z axis is perpendicular to the ground.

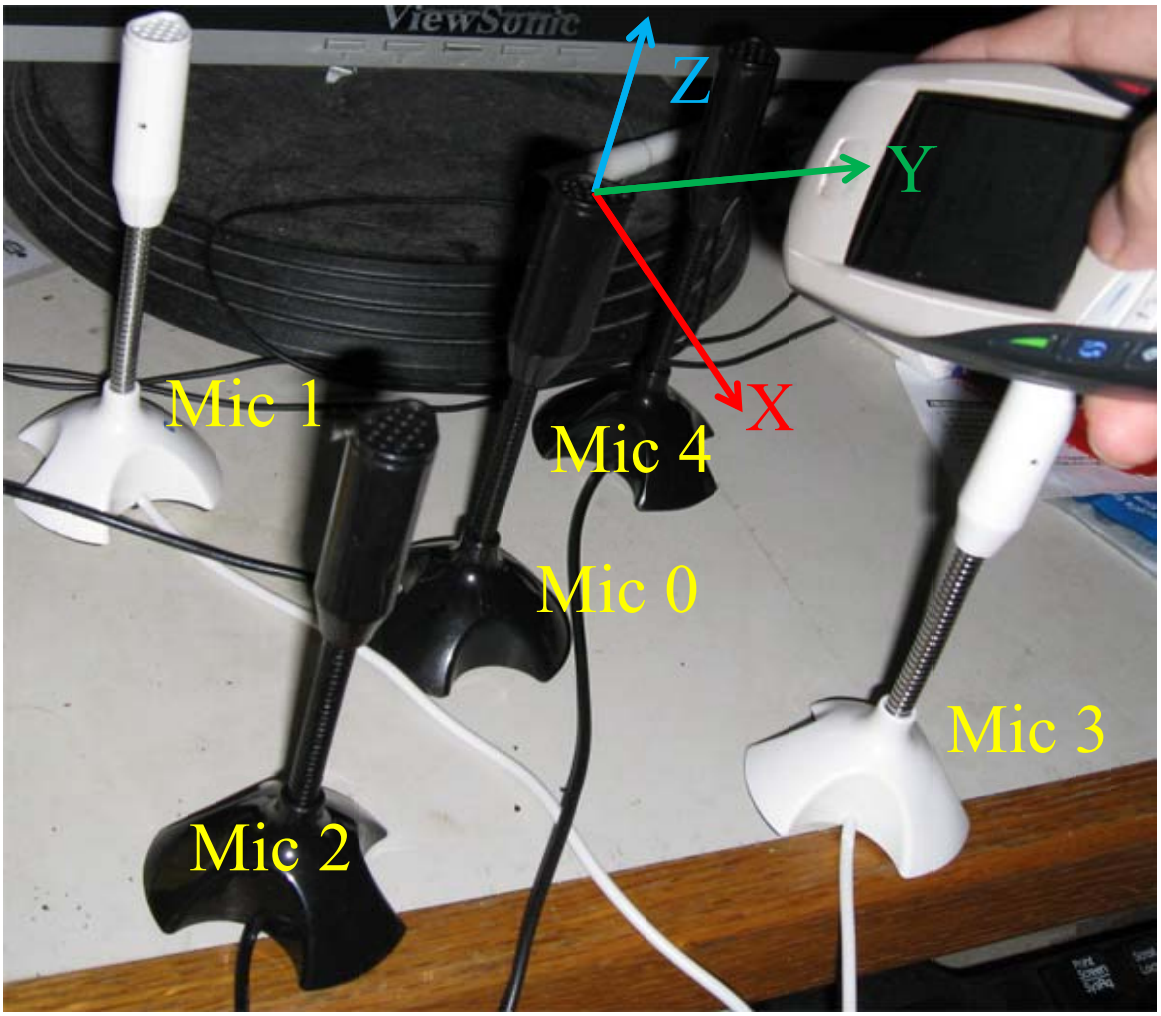

Figure 7.3: The physical position of the source [0.8, 0.6, 0.3]. The coordinates are rotated roughly 60 degrees anticlockwise referencing to **Figure 7.2**.

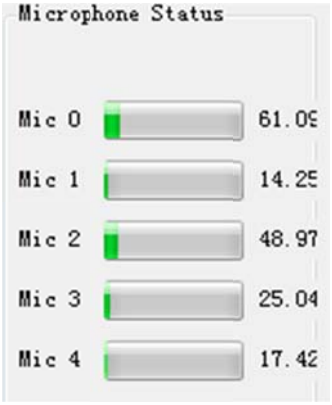

Figure 7.4: The corresponding microphone status (total energy received)

From **Figure 7.3** and **Figure 7.4** we can see that Mic 3 is closer to the source than Mic 0, but it receives a weaker signal. The reason is that Mic3 is positioned in a way that the side facing the phone receives the weakest response in its polar pattern.

The correspondence of **Figure 7.3** - **Figure 7.4** and **Figure 8.1** is shown in **Table 7.1**:

Label in Figure 7.2	**Label in Figure 8.1**
Mic 0	2
Mic 1	0
Mic 2	4
Mic 3	3
Mic 4	1

Table 7.1: Correspondence of Labels in Figure 7.3 and Figure 8.1

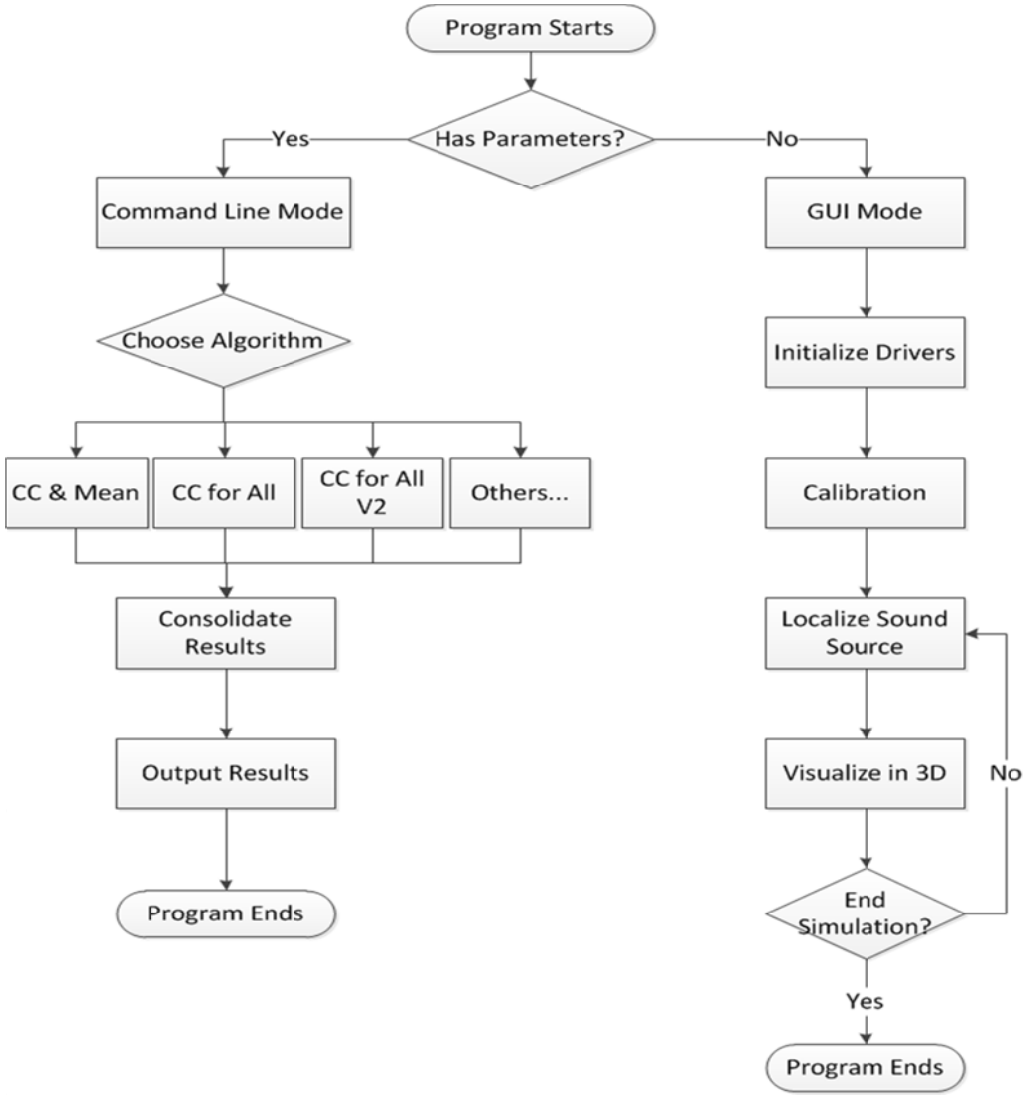

Figure 7.5: Flowchart of the software

The Command Line Mode branch takes stereo WAV files as input, and output the estimated delay between two channels for every 512 samples. It can also operate in batch mode, which will take multiple WAV files and output the results at the same time.

The GUI Mode branch operates in real-time. It can localize the sound source and render the coordinates onto the screen.

The Command Line Mode is extendible and can be extended by adding new algorithms.

The GUI Mode is scalable. It can adapt to 2D or 3D modes, and two to five microphones automatically. It can also be extended by adding different calibration and localization algorithms to the corresponding layer, because those layers take advantages of the façade design pattern.

CHAPTER 8

EXPERIMENTAL RESULTS OF RESOLUTION PROBLEM

Symbols and terms used in this chapter is listed in **Table 8.1**

Term/Symbol	Description
NS[x]	Number of samples needed to represent the signal in length x
L	The wavelength
NS[L]	Number of samples needed to represent the whole wavelength of the signal
Signal	All signals we use here are 1500Hz sine wave

Table 8.1: Symbols and terms in this chapter

The experiments are carried out in indoor conditions.

For TDE measurements, we first record 16 one-minute sound pieces using the equipment mentioned in the Section 7.1, and all experiments are performed based on those samples. Since there is no benchmark available for recorded data with physical devices, we are forced to compare the results between methods and those methods which can only reach integral resolution. We also make comparison with Giovanni's work using generated signals.

8.1. ENVIRONMENT SETUP

There are a few situations we want to highlight, in order to clearly describe the environment.

1. The microphones are inexpensive, so there is no guarantee that all microphones respond to the same to the same signal. In our experiments, we used five microphones to record 16 pieces of sound.

2. The room we used to record the sound pieces is a bedroom in a university apartment. There was some noise from the fans and the hard-disks inside a desktop computer, also high frequency buzz from the screen, which were largely negligible.
3. The microphones and the cell phone are placed on a hard desk, in front of a 26" – screen. This means we are expecting some reverberation effect.
4. The distance between two microphones is 50cm.

8.2. DATA COLLECTION

We first labeled the five microphones from 0 to 4, and we recorded the sound at the same distance for each of them. **Figure 8.1** shows the response for each of the microphones in waveform. There is no adjustment to amplitude.

The recording was done six times in six directions for each microphone. The recording with the largest sum of absolute amplitude was picked to generate **Figure 8.1**. The purpose of this was to remove the side effects caused by the polar pattern.

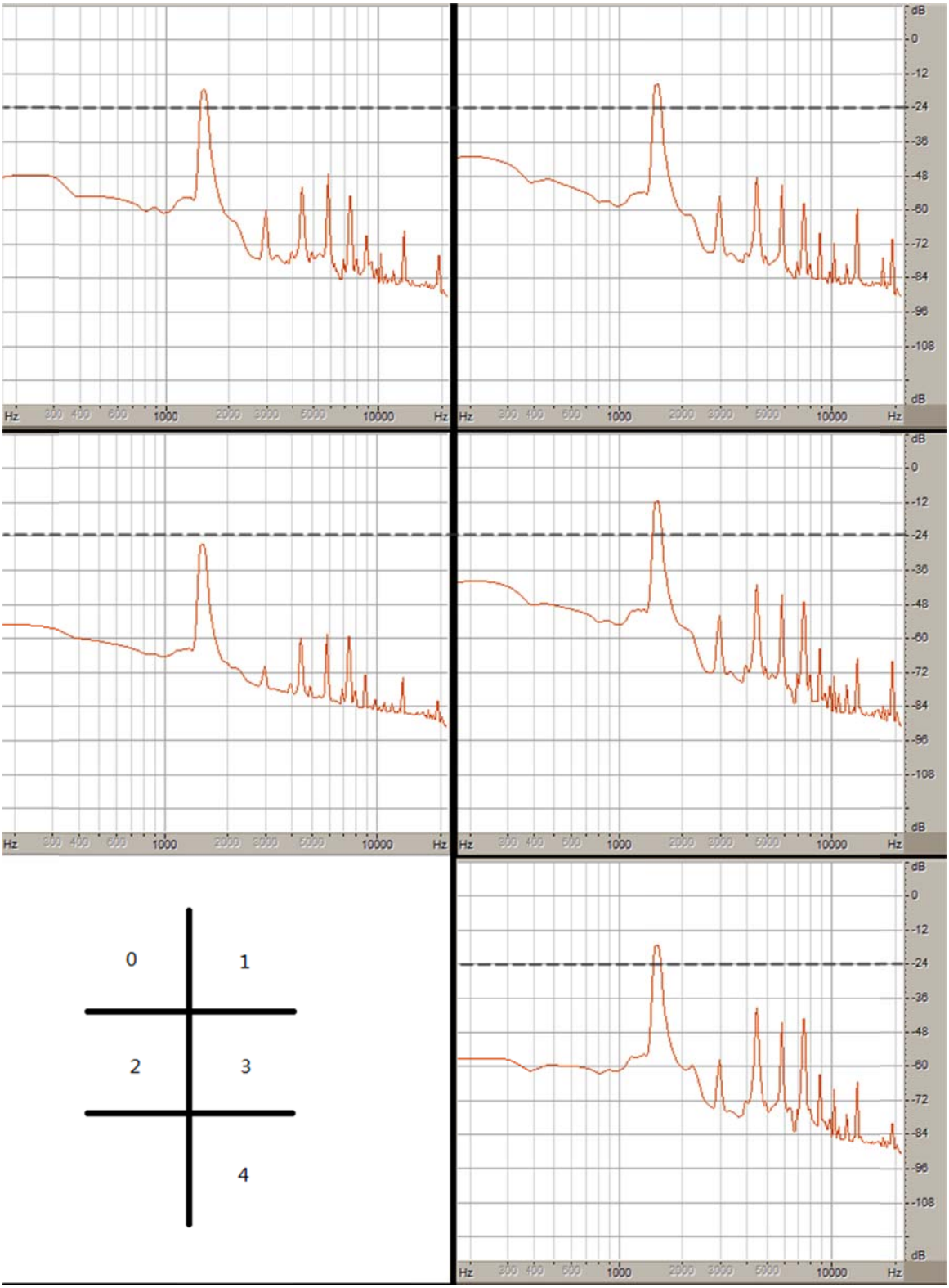

Figure 8.1: Response of different microphones to the same signal at the same distance and same noise level

By comparing the highest impulse to the reference line at -24dB of **Figure 8.1**, we are able to see that the number **2** microphone had an obviously weaker response to the 1500Hz frequency than the other microphones.

Table 8.2 summarizes how the data was collected:

No.	Microphones Used (A, B)	Distance to Microphone A (cm)	Distance to Microphone B (cm)	Noise Source	File Name (.wav)
1	0, 1	25	20	Natural	Normal_1_0
2	0, 1	20	25	Natural	Normal_0_1
3	1, 2	25	20	Natural	Normal_2_1
4	1, 2	20	25	Natural	Normal_1_2
5	2, 3	25	20	Natural	Normal_3_2
6	2, 3	20	25	Natural	Normal_2_3
7	3, 4	25	20	Natural	Normal_4_3
8	3, 4	20	25	Natural	Normal_3_4
9	0, 1	25	20	None	Silence_1_0
10	0, 1	20	25	None	Silence_0_1
11	1, 2	25	20	None	Silence_2_1
12	1, 2	20	25	None	Silence_1_2
13	2, 3	25	20	None	Silence_3_2
14	2, 3	20	25	None	Silence_2_3
15	3, 4	25	20	None	Silence_4_3
16	3, 4	20	25	None	Silence_3_4

Table 8.2: Data collection summary. There are two groups for comparison: Normal group (1 - 8) and Silence group (9 - 16)

The only difference between the normal group and the silent group was the existence of noise from the desktop computer and the screen. There was noise from the computer fans, and a high frequency buzz from the monitor screen. The computer was under a low workload, and the fan was spinning at 1800 rpm.

8.3. EVALUATION

We measured our performance by standard deviation, which is the evaluation method used in previous research, such as [8, 34, 35].

The reason we do not compare the result with a "ground truth" is that due to the size of the speaker and the microphone, it is **impossible** to define such "ground truth" which can reach the precision required (0.14 millimeter). The alternative is that we only measure the variation of the results. Less variation means more consistency, and better performance.

8.4. RESULTS

The raw outputs are in the form of "number of samples delay", meaning how many samples of signal A arrive earlier than that of B. We present only the variance of the raw data in this section, and give our analysis.

Figure 8.2 - Figure 8.5 present the variance computed from the samples collected. **Figure 8.6** compares results of Test 1 from the Noisy and the Silent groups.

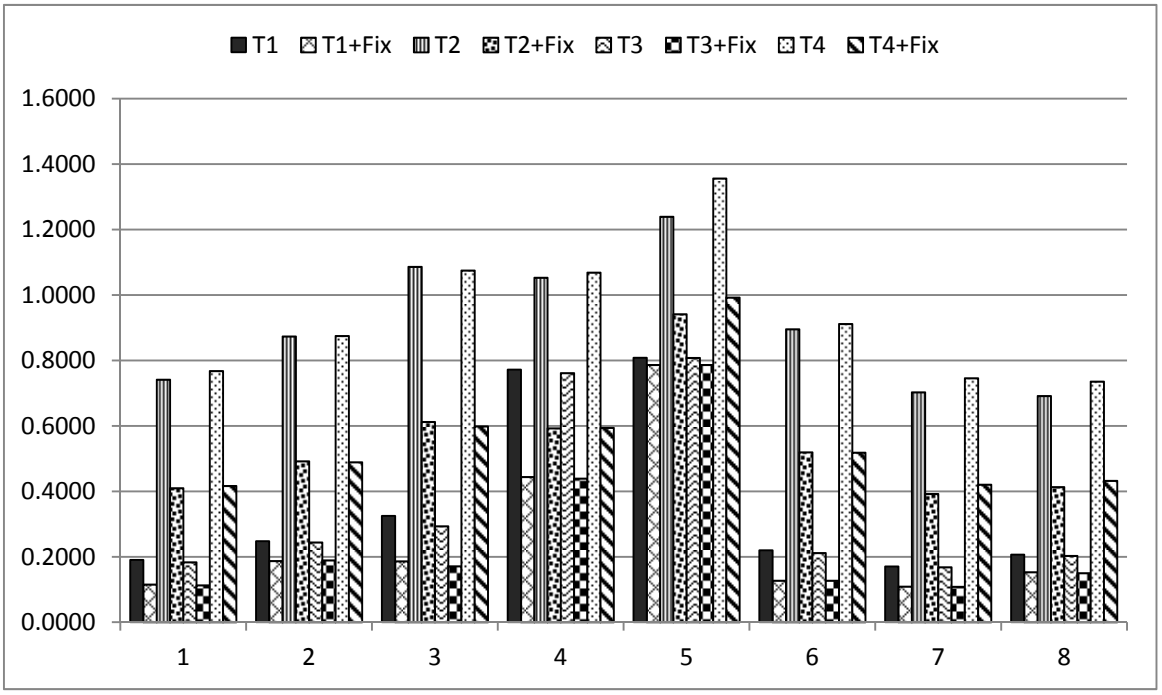

Figure 8.2: STDEV from Sample 1 - 8, "Tn" means Test n, "Tn + Fix" means adding smoothness fix to Test n

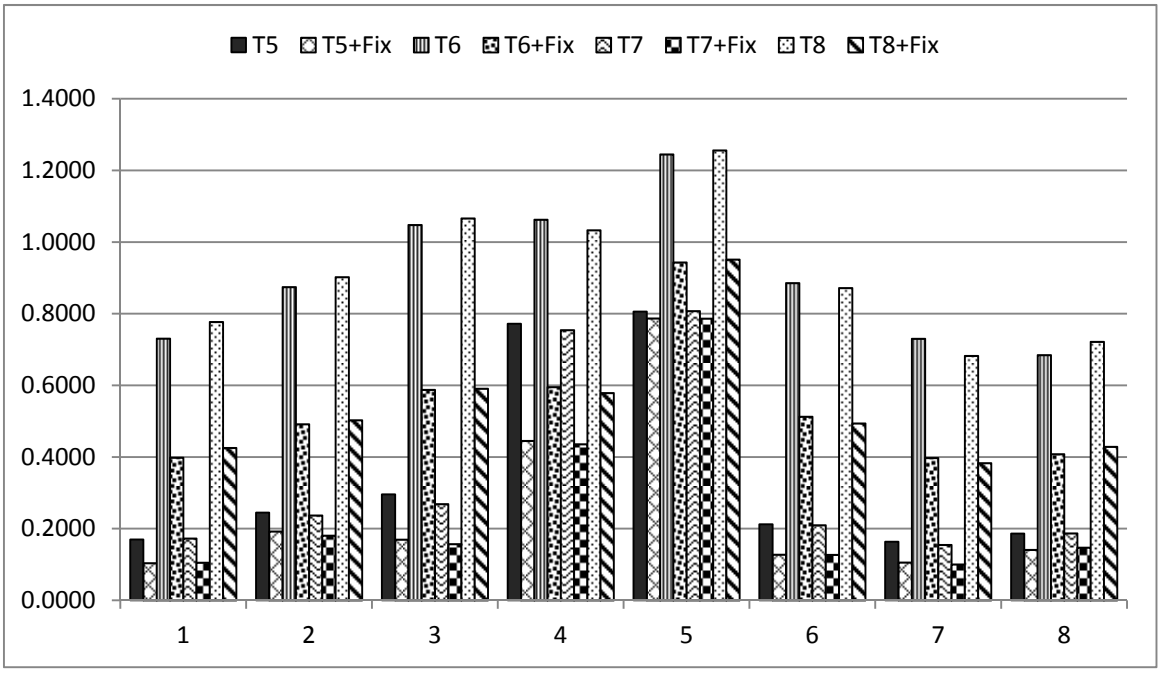

Figure 8.3: STDEV from Sample 1 – 8 (cont.), "Tn" means Test n, "Tn + Fix" means adding smoothness fix to Test n

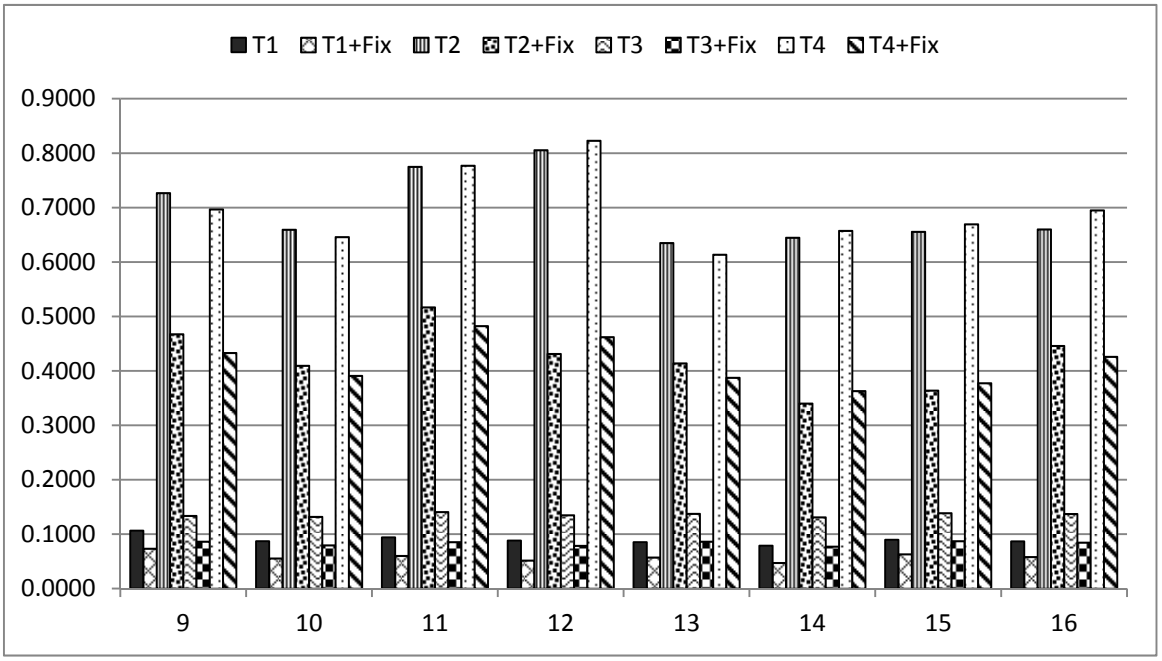

Figure 8.4: STDEV from Sample 9 - 16, "Tn" means Test n, "Tn + Fix" means adding smoothness fix to Test n

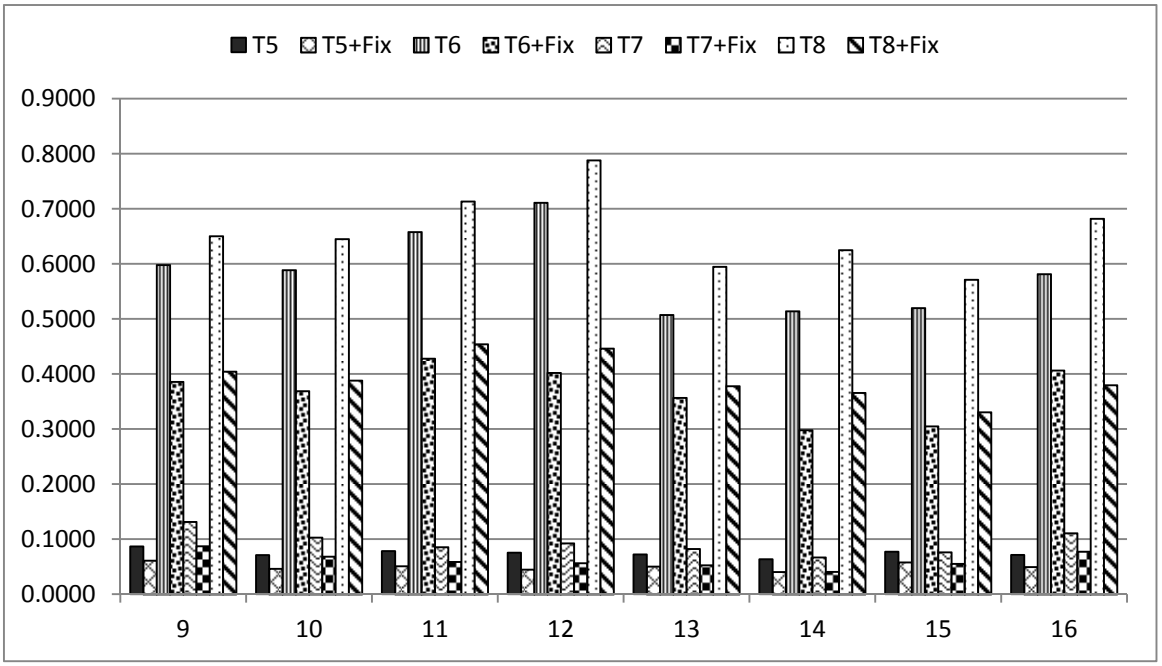

Figure 8.5: STDEV from Sample 9 – 16 (cont.), "Tn" means Test n, "Tn + Fix" means adding smoothness fix to Test n

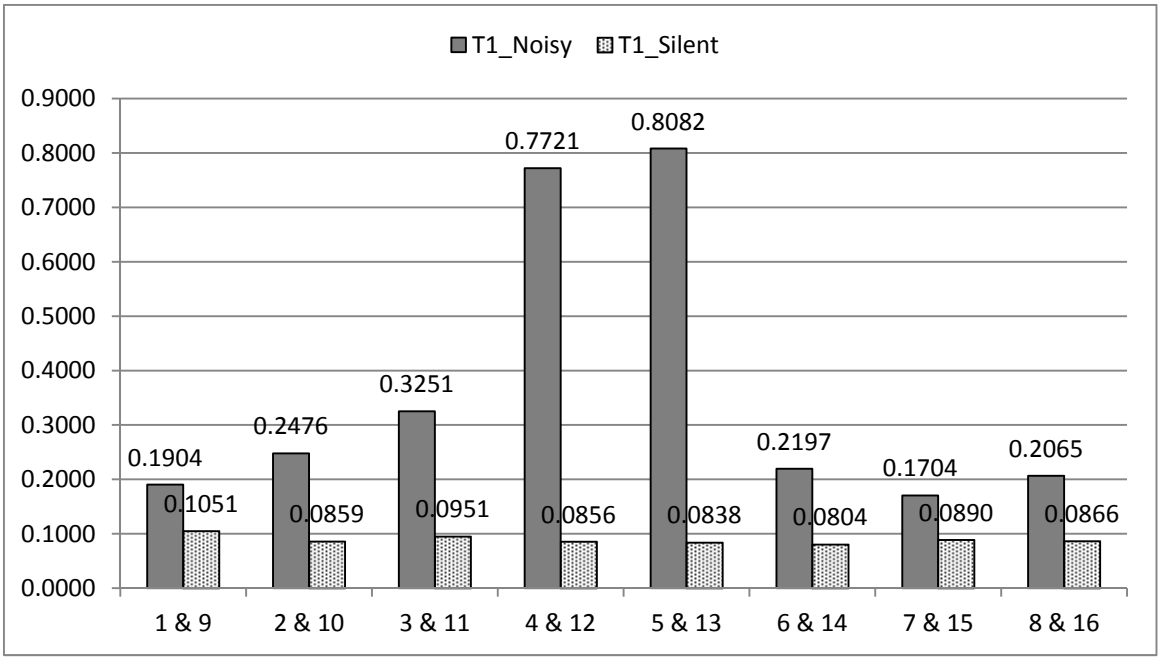

Figure 8.6: Comparison for T1 between samples with/without noise, "Tn" means Test n, "Tn + Fix" means adding smoothness fix to Test n

We also compare our results with Giovanni's work using generated signal. The signal we used is a generated sine wave of 1500Hz mixed with Gaussian white noise of variance 1 in terms of the value of samples, in [-1, 1]. We use ASDF as the cross correlation estimator, because this has best for both works. Results are compared in **Figure 8.7**.

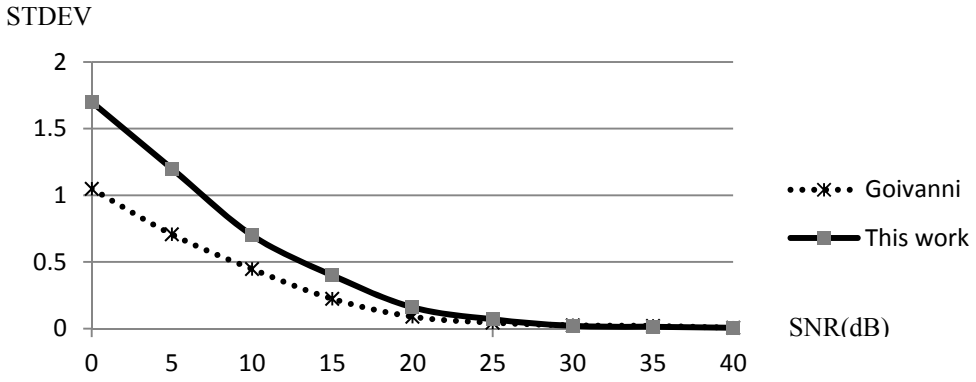

Figure 8.7: Comparison of this work and Giovanni's work using a generated sine wave mixed with white noise. The cross correlation estimator used is ASDF.

8.5. ANALYSIS

We can conclude from the results shown in **Figure 8.2 - Figure 8.6** that:

1. **Method**: **Test 1**, **Test 3**, **Test 5** and **Test 7** plus the **Smoothness Fix** yield smallest variance among the 16 combinations. The smallest variance is achieved by **Test 5** plus **Smoothness Fix** on test data number 14.
2. **Noise**: Noise poses a severe impact on the results. If we compare **Figure 8.2** & **Figure 8.3** with **Figure 8.4** & **Figure 8.5**, much smaller variances can be found in the latter group, which is silent group. We provide a clearer comparison only for **Test 1** in **Figure 8.6**.
3. **Hardware**: The response level of microphones to the specific frequency is clearly another key factor affecting the results. From **Figure 8.1** we know that microphone **2** does not respond as strongly at the 1500Hz frequency as others. If we look at **Figure 8.2** & **Figure 8.3**, Group 3 - 6, which are computed from noisy environments with one of the two channels provided by microphone **2**. The variances are higher than the others.
4. **Internal Factors**: If we compare the noisy group (**Figure 8.2** & **Figure 8.3**) with the silent group (**Figure 8.4** & **Figure 8.5)**, there is hardly any correlation between them. If we pay attention to the silent group only, we find some consistency in the results of the same method among different data sets. **Figure 8.6** shows clearly that when there is no audible noise, the variances of **Test 1** is far less when there is noise. The variances for silent groups are roughly at the same level. We can conclude that the

results from **Figure 8.4** & **Figure 8.5** are affected mainly by internal factors. They are: **1)** Quantization error of the sound card; **2)** Physical limitations of the microphone hardware (the hardware cannot respond precisely to the sound wave).

5. **Comparison**: The comparison shows that our method does not perform as well as Giovanni's while the signal is heavily corrupted by Gaussian white noise. The reason is that Giovanni's method takes the statistical information of the generated data (such as variance of noise) as inputs, so it performs better under noisy condition. While the SNR increase to 30 dB, our method starts to outperform Giovanni's.

Based on the analysis, we are able to conclude that those methods which tend to separate the data into segments, evaluate them individually and average the results, outperform those methods which tend to treat the data as a whole, and evaluate only once. The **Smoothness Fix** can further lower the variance, but it can be predicted that this "fix" will lower the responsiveness when we are tracking a moving sound source.

CHAPTER 9

CONCLUSION

In this thesis, we reviewed previous work related to sound source localization, including experimental and commercialized systems, and theoretical work. We also discussed classical models and algorithms used in the realm of TDE, which are the foundation of our work.

In this research, we explored two types of approaches to solve the sound source localization problem. The first is based on TDE methods, while the second is based on amplitude, or energy propagation.

We focused on the resolution problem in TDE, which is a classical problem but has not been studied intensively, especially under hardware limitations, as well as having the advantage of a controllable sound source. The results from our investigation are encouraging. Under room conditions with microphones of two USD each, the best standard deviation we achieved was 0.1002 (in number of samples) under noisy conditions; and 0.0406 under silent conditions. Assuming sound propagation speed is 330m/s, these numbers show that our system has the ability to give a measure that: **1)** there is probability of 70% that the measure is different from the true value for less than 0.075cm under room conditions, and 0.030cm under silent conditions; **2)** probability of 85% that the measure is different from the true value for less than 0.112cm under room conditions and 0.045cm under silent conditions. These results did not meet our requirement exactly (0.014cm with 85% probability).

While we are comparing this method with Giovanni's method on generated signals, the goal can be reached when SNR is greater than 35dB for both methods. It gives us the confidence that if we can improve the quality of sound source and the sensor within the budget, we are able to get even better results for real data.

As a separate stream, we also prototyped the energy propagation based approach. Its performance was not acceptable, for the following reasons: **1)** The model we used was too simple to capture all the energy propagation features and hardware characteristics, such as reverberation, and the polar pattern of the microphone response. **2)** The blocking effect caused by the operator(s) is neither preventable nor predictable. Therefore, there is not likely to be a proper model for accurate measurement. Moreover, having complicated models will dramatically increase the required computational power, affecting the ability of "real-time evaluation".

Any future studies would concentrate on TDE based methods because they are not affected by the blocking effect of the operator as severely as energy propagation based methods. This gives the system more freedom, meaning a wider area of application. Also, to achieve real-time operation the algorithm is highly distributable; the methods we tested are potentially distributable, because the computation of cross correlation is distributable. Possible improvements to the system will be using ultrasound waves to replace audible sound sources, which are quiet, and far more resistant to environmental noise. The sensors should also be replaced with ultrasonic pickup devices.

REFERENCES

[1] "Personal Mobility & Manipulation Appliance", Carnegie Mellon University http://www.cmu.edu/qolt/Research/projects/permma.html Accessed March 2011.

[2] Kyo Hoshino, "Fast & Accurate Gesture Driven Interface", Hoshino Lab - Tsukuba University http://exponet.nikkeibp.co.jp/ij2010/exhibitor/view/78 Accessed March 2011; Diginfo news "Gesture-Driven Robot Arm System" http://www.youtube.com/watch?v=UjbZYN1Db14 Accessed March 2011.

[3] ALEX BOERNER, "Partial knee reconstruction surgery" http://www.tcpalm.com/photos/2010/feb/10/231252/ Accessed March 2011.

[4] "MAKOplasty@ - The Patient Specific Robotic Arm System for Knee Arthroplasty", MAKO Surgical Corp, http://www.makosurgical.com/makoplasty/ Accessed March 2011.

[5] Acoustic Magic, http://www.acousticmagic.com/ Accessed March 2011.

[6] J. E. Thorner, "Approaches to Sonar Beamforming", in Proceeding IEEE Southern Tier Tech. Conference., pp. 69-78, Apr. 1990.

[7] L. Kleeman and R. Kuc, "An optimal sonar array for target localization and classification", Proceeding IEEE International Conference on Robotics and Automation, vol.4, pp. 3130-3135, 1994.

[8] Michael S. Brandstein, Harvey F. Silverman, "A practical methodology for speech source localization with microphone arrays," Computer Speech & Language, Vol.11, no.2, April 1997, Pages 91-126, 1996.

[9] K. Nakadai, H.G. Okuno, and H. Kitano, "Real-time multiple speaker tracking by multi-modal integration for mobile robots", Proceeding of EUROSPEECH 2001, pp. 1193-1196, 2001.

[10] K. Nakadai, H.G. Okuno, and H. Kitano, "Real-time sound source localization and separation for robot audition", In Proceedings of 2002 International Conference on Spoken Language Processing (ICSLP-2002), pp. 193-196, 2002.

[11] V.M. Trifa, A. Koene, J. Moren, and G. Cheng, "Real-time acoustic source localization in noisy environments for human-robot multimodal interaction," Robot and Human interactive Comm., pp. 393-398, Aug. 2007.

[12] Knapp, C.; Carter, G., "The generalized correlation method for estimation of time delay," IEEE Transactions on Acoustics, Speech and Signal Processing, vol.24, no.4, pp. 320- 327, Aug 1976.

[13] R. Moddemeijer, "An information theoretical delay estimator", University of Twente, Enschende (NL), Technical Report: 080.87.45, 1987.

[14] Baumgarte, F., "Improved audio coding using a psychoacoustic model based on a cochlear filter bank," IEEE Transactions on Speech and Audio Processing, vol.10, no.7, pp. 495-503, Oct 2002.

[15] J.-M. Valin, F. Michaud, J. Rouat, D. Letourneau, "Robust sound source localization using a microphone array on a mobile robot". Proceeding 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003(IROS 2003). vol.2, pp. 1228-1233 vol.2, 27-31 Oct. 2003.

[16] Han Yi; Wu Chu-na, "A new moving sound source localization method based on the time difference of arrival," 2010 International Conference on Image Analysis and Signal Processing (IASP), pp. 118-122, 9-11 April 2010.

[17] "DUO pen", Pan and Free Co. Ltd, http://www.penandfree.com/ Accessed March 2011.

[18] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," IEEE Transactions on Signal Processing, vol.42, no.8, pp. 1905-1915, August 1994.

[19] Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang, "Time Delay Estimation in Room Acoustic Environments: An Overview", EURASIP Journal on Applied Signal Processing, pp.1-19, 2006.

[20] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," Journal of the Acoustical Society of America, vol.107, no.1, pp. 384–391, 2000.

[21] A. Stephenne and B. Champagne, "Cepstral prefiltering for time delay estimation in reverberant environments", in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’95), vol.5, pp. 3055-3058, Detroit, Mich, USA, May 1995.

[22] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms", in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’97), vol.1, pp. 375–378, Munich, Germany, April 1997.

[23] C. S. Clay and H.Medwin, "Acoustical Oceanography", New York, NY, USA, 1977.

[24] Jacovitti, G.; Scarano, G., "Discrete time techniques for time delay estimation," IEEE Transactions on Signal Processing, vol.41, no.2, pp.525-533, Feb 1993.

[25] F. A. Reed, P. L. Feintuch, and N. J. Bershad, "Time delay estimation using the LMS adaptive filter--static behavior," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.29, no.3, pp. 561-571, 1981.

[26] Nishiura, T.; Yamada, T.; Nakamura, S.; Shikano, K., "Localization of multiple sound sources based on a CSP analysis with a microphone array, "IEEE International Conference on Acoustics, Speech, and Signal Processing 2000 (ICASSP '00), vol.2, pp.II1053-II1056, 2000.

[27] J. Chen, J. Benesty, and Y. A. Huang, "Robust time delay estimation exploiting redundancy among multiple microphones," IEEE Transactions on Speech and Audio Processing, vol.11, no.6, pp. 549-557, 2003.

[28] Y. (Arden) Huang and J. Benesty, "Adaptive multichannel time delay estimation based on blind system identification for acoustic source localization," in Adaptive Signal Processing-- Applications to Real-World Problems, J. Benesty and Y. (Arden) Huang, Eds., chapter 8, pp. 227-248, Springer, Berlin, Germany, 2003.

[29] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay", IEEE Signal Processing Mag., vol.13, pp. 30-60, 1996.

[30] D. L. Maskell and G. S. Woods, "The discrete-time quadrature subsample estimation of delay", IEEE Transactions on Instrumentation and Measurement., vol.51, no.1, pp. 133-137, 2002.

[31] The Hong Kong Golden Computer Market, http://www.hkgolden.com/ Accessed March 2011.

[32] Markheath, "NAudio", http://naudio.codeplex.com/ Accessed March 2011.

[33] "SlimDX", http://slimdx.org/ Accessed March 2011.

[34] L. Kleeman and R. Kuc, "Mobile robot sensor for target localization and classification", The International Journal of Robotics Research, vol.14, no.4, pp. 295-318, 1995.

[35] M.S. Brandstein, J.E. Adcock, and H.F. Silverman, "A closed-form location estimator for use with room environment microphone arrays", IEEE Transactions Speech Audio Processing, vol.5, pp. 45-50, 1997.