Audio Based Location Fingerprinting for Mobile Devices

By

Nikhil Suryanarayanan

Wisc ID: 9057147812

Project Report in partial fulfillment for the degree of

Master of Science

Advisor: Prof. Parmesh Ramanathan

Dept. of Electrical & Computer Engineering

University of Wisconsin – Madison

Fall 2010

INDEX

I Introduction

The main goal of the project is to enhance the context-awareness of cellular handsets by exploiting commonly available technologies/capabilities present on the devices. For instance, most cellular handsets today come with a variety of sensors and capabilities including location sensors, accelerometers, light sensors, cameras and high-fidelity speakers and microphones. By fusing information gathered from multiple sensors, cellular handsets can tailor their services based on the user's context.

Location is a major component of user's context. In open outdoor environments, Global Positioning System (GPS) can provide the needed location information to infer user's context. However, location information is not easy to obtain when the user is either indoors or in obstruction-laden outdoor terrain. Our goal in this project is to exploit other resources in cellular handsets to improve the location context of users in environments where GPS is not readily available. GPS is currently the most precise estimation of location.

Wi-Fi and Cell-Tower based location services are also currently available. They help determine the approximate location to the accuracy of approximately 100 meters. This is not very helpful when if we need to determine the user context within a building likes office or homes. The need to precisely determine location in an indoor environment by exploring readily available information from the surrounding is the main idea of the project.

There can be several uses of trying to estimate location indoors. The mobile devices can switch user modes when the user moves across a building from office suite to a conference suite, without the intervention of the user. Also other features like automatically connecting to wireless networks or syncing up emails to the local server in the building are some of the compelling advantages.

II Acoustic Location Estimation

Specifically, the mobile device's audio system (i.e. speaker and microphone) is used to identify the user's location from a set of possible choices. There is a potential for widespread use of this technique as primarily all devices are equipped with this basic functionality.

The basic idea of our approach is as follows. A software running in the mobile device is triggers a periodic routine that the phone's speaker and microphone performs. The device emits short sequences of audio chirps and records the multipath reflections or in other words the time delayed version of the original signal from the various surfaces in the user's environment. Using signal processing techniques, the multipath Time Delay Estimates (TDEs) from the recorded reflections are computed. The vector of TDEs is then used to estimate the most likely location of the user given prior information from other sensory information from other sensory inputs. By fusing this audio based information with similar information from other sensory inputs, the probability of correctly estimating the user's location is expected to be much higher than currently possible.

For example, supposed that a cellular handset user wants a certain service response when he/she is in or around her office desk. Further suppose that the TDE signature of the corresponding location has been pre-characterized using either a user-triggered or self-triggered training sequence. By comparing this signature with real-time TDE estimates, the software will recognize when the user is near his/her desk in order to activate the desired service response.

Unlike other positioning systems like ultrasound or infrared/infrasonic, the user is not required to carry and additional or specialized hardware. The system has virtually no additional cost and can be deployed in a widespread manner across all mobile device platforms with ease. Thus the system is universally compatible.

Some of the other techniques used for indoor location sensing deploy a combination of radio frequencies and ultrasound for indoor positioning. These techniques require the environment to have pre-installed transmitters. By looking up the frequency of the transmitted signal or by comparing the intensity of the received audio signal, location can be determined. However, these techniques are not suitable for indoor positioning in public places especially for ordinary users. Most of these systems need the use of specialized hardware which is not easily available. Also some of these systems are prohibitively expensive for wide deployment.

III Sound Reflection Model

Figure 1 shows the general model of sound reflection in a room. The audio chirp transmitted from the user's device will follows a similar model and encounter multiple reflections. These reflections will appear as time delayed versions of the transmitted signal in the recorded waveform. Depending on the size of the room and attenuation of the sound waves at the reflecting surfaces, one or more of the possible reflecting path might be attenuated.

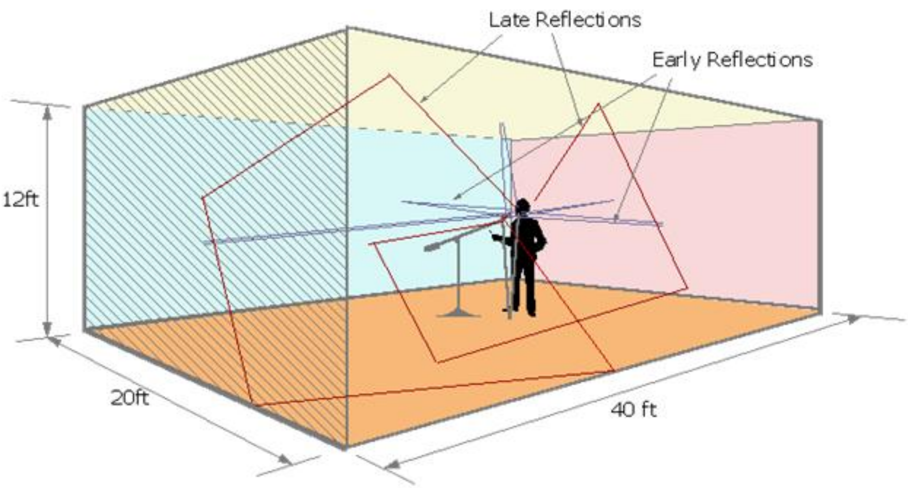


***Figure 1**: Reflection Model in an Indoor Environment*

The location estimation would primarily have two phases. The first phase can be attributed as the learning phase while the second phase can be called the detection phase. In the learning phase, the user can train the phone to listen to the reflection pattern of the room that he or she is currently in. The process involves transmission of short audio chirp. The software on the phone will be also recording at the same time. The audio recording will contain self-recording of the transmitted signal and any time-delayed versions of the transmitted signals, if present. A feature vector will be created using the methods described later in this report. The feature vector is actually a representation of the times that the TDEs of the transmitted signal were observed. Similar operation is performed at all location where the user desires the location be detected. Thus a database of feature vectors will be created.

During the detection phase, an audio chirp of similar or smaller length will be transmitted. Again the software will record the audio waveform during this period. The TDEs are calculated using exactly the same method as done during the learning phase. The resulting vector which represents the time points at which the time-delayed version was observed will be compared against the database. A classification algorithm like nearest-neighbor would be able to predict the closest match of the vector generated during the detection phase with the features vectors that are stored in the phone's database. If a close match is obtained the location is displayed. The location detection can be translated into some user-desired routine like switching phone modes or connecting to a different access point etc.

Figure 2 shows an ideal example of how the reflections from the room will look. The dominant wave is the self-recording while the subsequent waves are the time-delayed versions of the transmitted signal. The below wave was synthetically generated for demonstration. Real-world examples might not contain time-delayed signals that can be distinguished by naked eye. Thus sophisticated signal processing algorithms might have to be used to extract these signals as they might be almost distinguishable from the noise levels in the received signal.

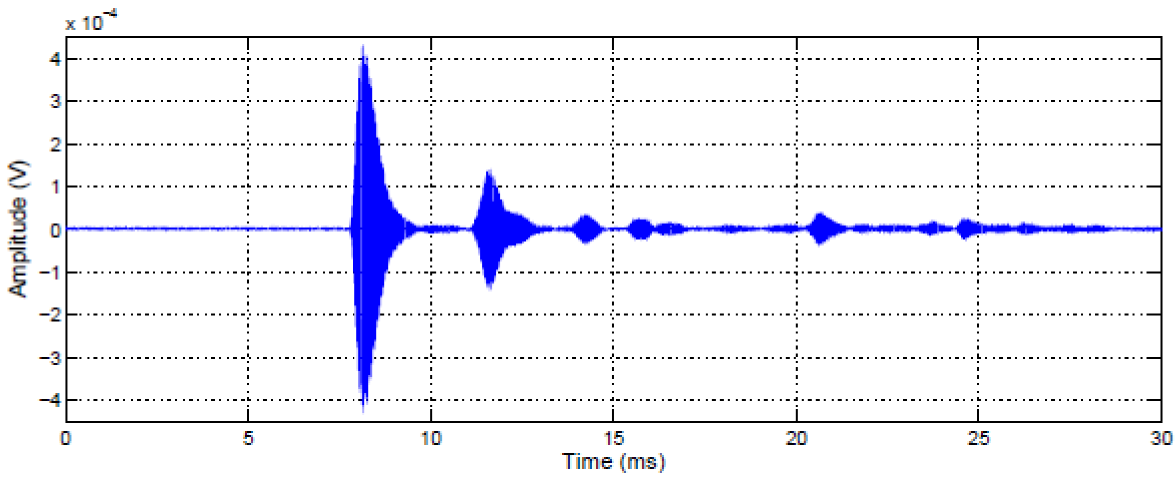


Figure 2*: Received Pulse containing reflections*

IV Methods to Detect Time Delayed Signals

There are several methods to detect Time Delayed version of the original signal. They can be either time-domain based or frequency domain based. Some of the time domain methods include Auto-Correlation, Golay codes etc., while frequency domain techniques include Short Time Fourier Transform (STFT) & Weiner Filter.

The methods will be described below with synthetic and practical waveforms for each methodology.

1. **Weiner Filter [2]**

Weiner filter is an unconstrained optimal filter and requires no requirement for prior knowledge of noise. The typical application of Weiner filter is filtering or deconvolution. Weiner filter can be applied in estimating audio chirp echo time delay and the estimated time delay can be used to create the feature vector.

The model of linear time-invariant system can represented as shown below in Figure 3.

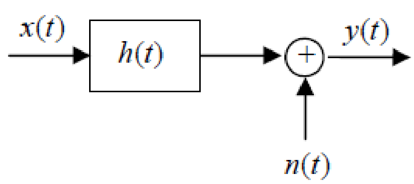


Figure 3: *Model of a linear time-invariant system*

Here x(t) is the system input, h(t) is the system impulse response, n(t) is additive noise, y(t) is system output. Then the system mathematical model is

$$y(t) = x(t) * h(t) + n(t) \ldots\ldots.. \text{(i)}$$

Here * is the convolution operator. The system model in the frequency domain is

$$Y(\omega) = X(\omega)H(\omega) + N(\omega) \ldots\ldots\ldots \text{(ii)}$$

When the system and its output are known, the optimal estimation of system under certain criterion can be achieved by certain technology, this processing is commonly called inversion or deconvolution. Deconvolution is commonly realized by filtering. Figure 4 represents the deconvolution model in frequency domain, here G(ω) is the deconvolution filter and output $\hat{X}(\omega)$ is the estimation of X(ω) . Then their mathematical relation is $\widehat{X}(\omega) = Y(\omega)G(\omega)$.

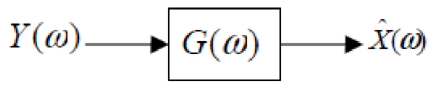


Figure 4: *The frequency domain model of deconvolution*

Weiner filter is an optimal filter under minimum mean square error (MMSE). It can be used to deconvolute Y(ω) and the result is the optimal estimation of X(ω) under MMSE.

a. Echo-Time Delay Estimation Based on Weiner Filter

We can build a simple model of detecting echo time delays. If x(t) is the emitted audio chirp, h(t) is the transmitting system, n(t) is the noise and the output is represented as y(t) . If s(t) is original signal, the received signal can ideally modeled as in Eq. iii.

$$y(t) = a_1 s(t - t_1) + a_2 s(t - t_2) + a_3 (t - t_3) + n(t) \; \ldots \ldots (iii)$$

Here a_1, a_2 & a_3 are the attenuation coefficients and t_1, t_2 & t_3 are the time delays of the echoes of the original signal s(t) . The Fourier Transform of (iii) is

$$Y(\omega) = a_1 S(\omega) e^{-j\omega t_1} + a_s S(\omega) e^{-j\omega t_2} + a_3 S(\omega) e^{-j\omega t_3} + N(\omega) \ldots \ldots (iv)$$

$$Y(\omega) = B(\omega) S(\omega) + N(\omega) \ldots \ldots (v)$$

$$\hat{B}(\omega) = Y(\omega) G(\omega) = Y(\omega) \frac{S^*(\omega)}{K + |S(\omega)|^2} \ldots \ldots (vi)$$

The inverse Fourier transform will be represented as

$$\hat{b}(t) = a_1 \delta(t - t_1) + a_2 \delta(t - t_2) + a_3 \delta(t - t_3) \ldots \ldots (viii)$$

Obviously, there are three impulses at $t_1, t_{2,} t_3$ in $\hat{b}(t)$. Therefore, the time delay estimation will be achieved by searching for the strong impulse in $\hat{b}(t)$. These peaks can be used to create a feature vector.

b. Explanation using Waveforms

The figures below represent synthetically generated waveforms. The waveforms contain noise and reflected signals of decreasing amplitude as is observed in a general case. The peaks point to the existence of the time-delayed copy of the transmitted signal. A simple peak detection algorithm can be used to pick out the peaks. This can be used to create the feature vector that was discussed earlier.

In the absence of noise, the recorded waveform is as shown in Figure 5

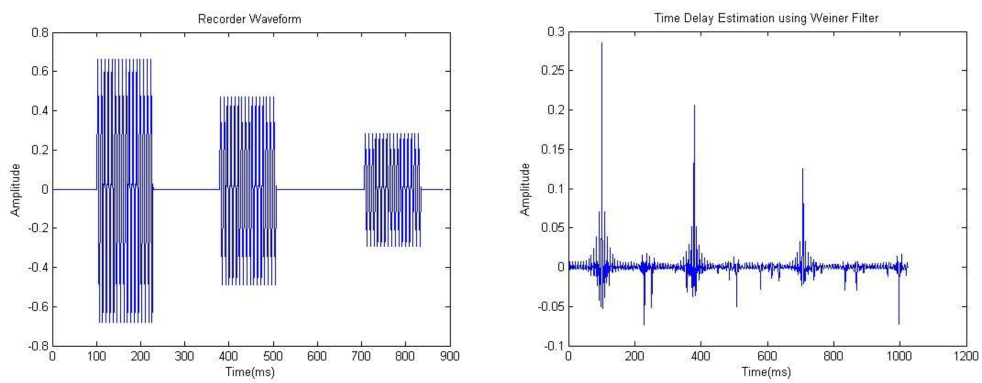


Figure 5*: Recorder Waveform and Pulse Detection using Weiner Filter*

In the absence of noise

.

It can be observed that the amplitude of the received signal determines the amplitude of the peak. If the amplitude of the time-delayed signal is very low, the subsequent amplitude of the corresponding peak will be low. This can be observed in the following figure. If the noise levels comparable to signal levels, the time-delayed samples might be impossible to recover.

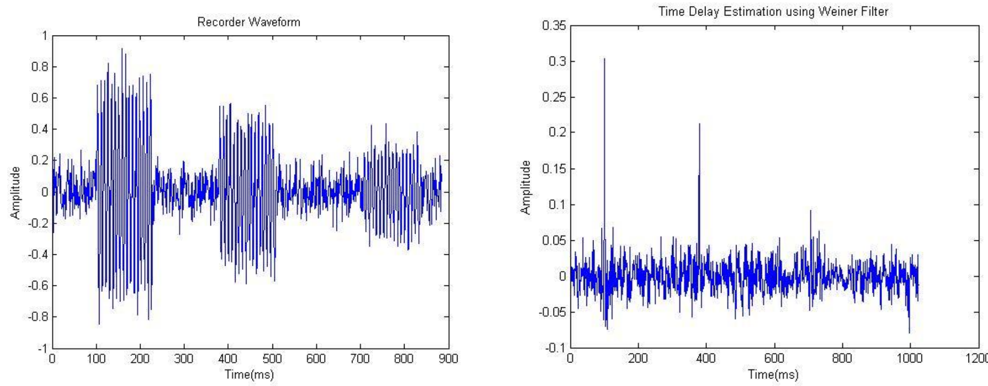


Figure 6*: Recorder Waveform and Pulse Detection using Weiner Filter*

Noise levels are comparable to signal

Thus if the noise levels are low, Weiner filter method of time-delay estimation is an effective method. Autocorrelation of the received signal with the transmitted signal is another useful method. But this can be computationally intensive as we have to convolve the received signal at each time point to check if the copy of the received signal exists. Also the peaks will not be as distinguishable as the in the above case. A simple case of convolving rectangular pulse with itself will result in a triangular pulse.

Multipath can create diversity and the received signal might not have two or more time-delayed versions arriving at the same time. A similar case is analyzed in the example below. Weiner filter is used to identify the echoes of the transmitted signal. The example uses the same signals as used in the case with zero noise.

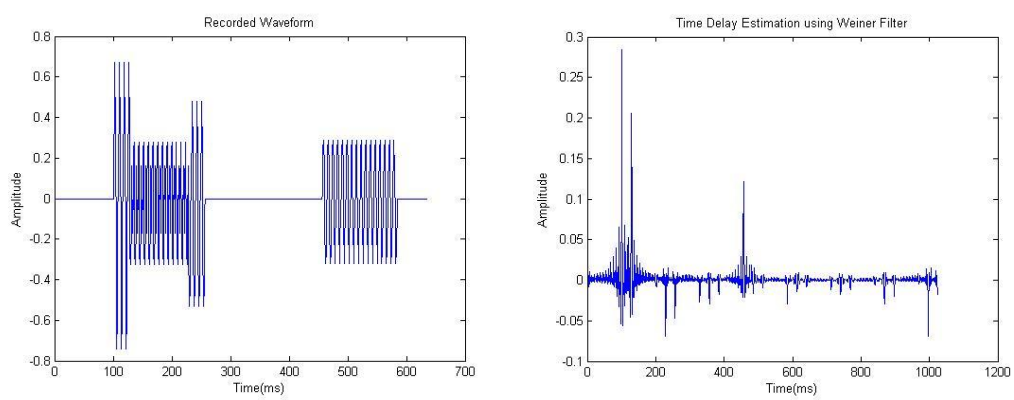


Figure 7*: Recorder Waveform and Pulse Detection using Weiner Filter*

Overlapping pulses, zero noise

It can be observed that even in the presence of diversity, the time delayed versions of the signals can be easily detected. In the presence of noise also, the technique provides performance similar to the case when the time-delayed chirps are not overlapping.

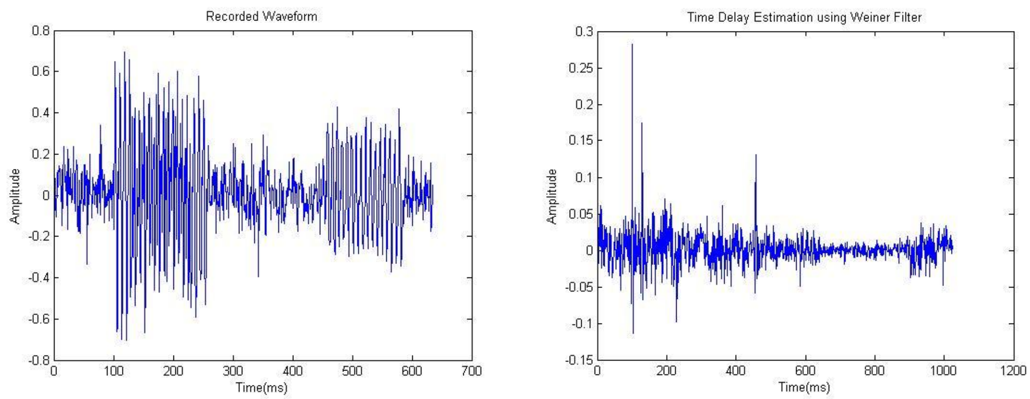


***Figure 8**: Recorder Waveform and Pulse Detection using Weiner Filter*

Overlapping pulses, in presence of noise

2. Short Time Fourier Transform (STFT) [5]

Short Time Fourier Transform (STFT) is a Fourier Related Transform used to determine the sinusoidal frequency and phase content of the local sections of a signal as it changes over time. It has been observed during the course of this project that in several cases the reflected signals do not retain the shape or characteristics of the transmitted waveform. The reflected wave pattern might only contain the dominant copy of transmitted frequency at certain points of time in the recorded waveform.

STFT is effective in such cases as the frequency might not be present over the entire range and a simple FFT might not indicate the presence of the desired frequency. But if the Fourier transform was calculated for short windows of the received signal, the transmitted frequency is revealed. STFT by itself cannot be used to create the feature vector as the Window size and other computational parameters will determine how much accuracy can be achieved. It is not computationally feasible to compute STFT at each time points with a given window size.

We can use the STFT to narrow down on sections of the recorded waveform which might contain the time delayed version of the transmitted pulse. After which, other time-domain methods can be used to check for the presence of signal. An illustrative example is shown in the Figure 9

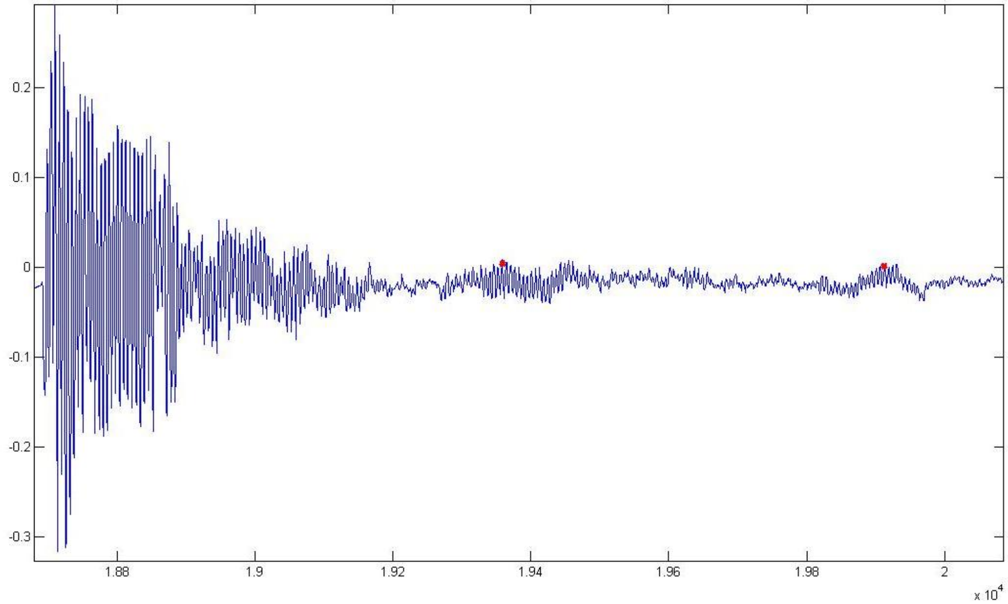


***Figure 9**: Recorded waveform shows presence of different frequency.*
The time points are marked in red

As seen in the figure above, the points marked in red, indicate the time points at which the reflected signals exist.

3. Golay Complementary sequences [1]

Golay complementary sequences are used in the time-domain analysis to detect the presence of the time-delayed signal. It is more robust to additive white noise. The length ***L*** bi-level sequences a(n) and b(n) are Golay Complementary sequences if and only if the following condition holds, where * denotes the autocorrelation operator.

$$a(n) * a(n) + b(n) * b(n) = 2L\delta(n) \ldots \ldots (ix)$$

Where $\delta(n)$, denotes the Kronecker delta function. Many references in the audio signal processing literature refer to such sequences as Golay codes. Given that $a_L(n)$ and $b_L(n)$ are Golay, it turns out that $a_{2L}(n) = [a_L(n) \quad b_L(n)]$ and $b_{2L}(n) = [a(n) \quad -b_L(n)]$ are also Golay.

Sequences like $a = [1 \quad 1]$ and $b = [1 \quad -1]$ are used to create the Golay complementary sequence.

To measure the impulse response of the system under test, sequence '**a'** is transmitted and then sequence '**b'** is transmitted. The response to both the sequences is recorded. Let $r_a(n) = a(n) * h(n)$ be the response due to input and $r_b(n) = b(n) * h(n)$ be the response due to input b respectively. The impulse response of the system is measured as follows

$$h(n) = \frac{1}{2L}\big(a(n) * r_a(n) + b(n) * r_b(n)\big) \ldots\ldots (x)$$

The above technique can be explained as follows. Let's assume that that we choose Golay code of length N = 2. Thus the Golay sequence will have length 2^N = 4. Thus the Golay complementary sequences will looks as shown below

a = [1   1   1   -1];
b = [1   1 -1   1];

The autocorrelation functions of the above two sequences are (4,1,0,-1) and (4,1,0,1), which adds up to (8,0,0,0). Thus If **a** & **b** were the recordings due to their respective transmitted sequences. By autocorrelating and adding the resultant vectors, we can check if the signal actually existed. If the sequence was not actually that of a and/or b then the resulting vector will not have such high value in its first index and other indices will have non-zero values. It is observed that the amplitude of the resulting vector is 8, while the recorder sequence had only amplitude of 1. This is the key feature of Golay Complementary sequence.

The code amplifies the existence of the original signal. Hence in presence of noise, the signal can be effectively recovered. The gain is 2*(2^N). Longer the length of the Golay complementary sequence, higher will be the amplification obtained and subsequently signal buried deep in noise can be effectively recovered and echoes can be detected.

a. Time Delay Estimation using Weiner Filter

As stated above, both the complementary sequences are transmitted. The time-delayed signals for both the sequences will be at the same time points. After Autocorrelating the recorded waveforms with their respective transmitted sequences, the vector resulting from their addition will help in creating the feature vector. The time-points where the reflected/time-delayed copy of the transmitted audio chirp will have peaks. A peak detection algorithm similar which is used for Weiner Filter method can be used to detect the time-points in the recorded waveform.

b. Explanation using Waveforms

In the absence of noise, the length of the Golay sequence is not of much consequence. The time-delayed signals can be easily extracted in a clean way. Figure 10 below shows an example of estimating the presence of reflected/time-delayed versions of the signal.

In the absence of noise,

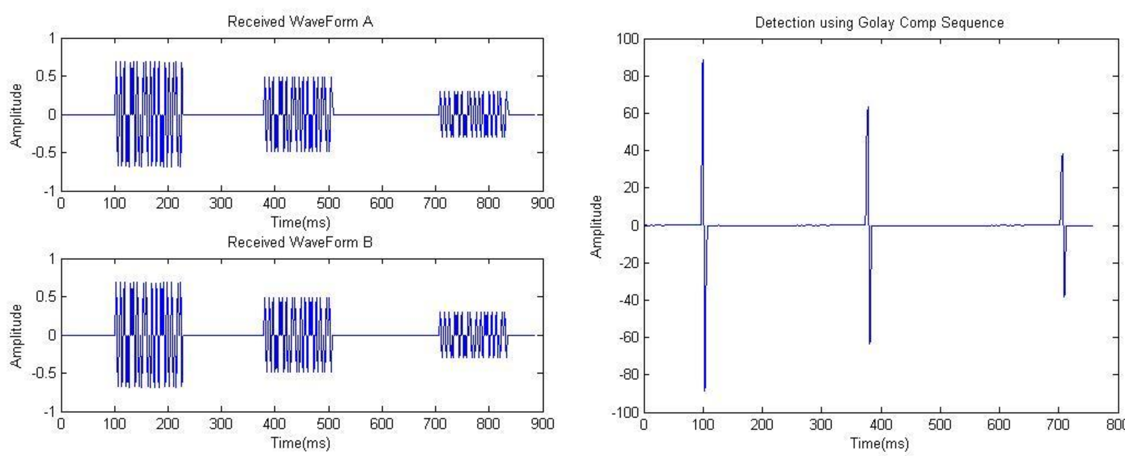


Figure 10*: Length N = 4. Zero Noise*

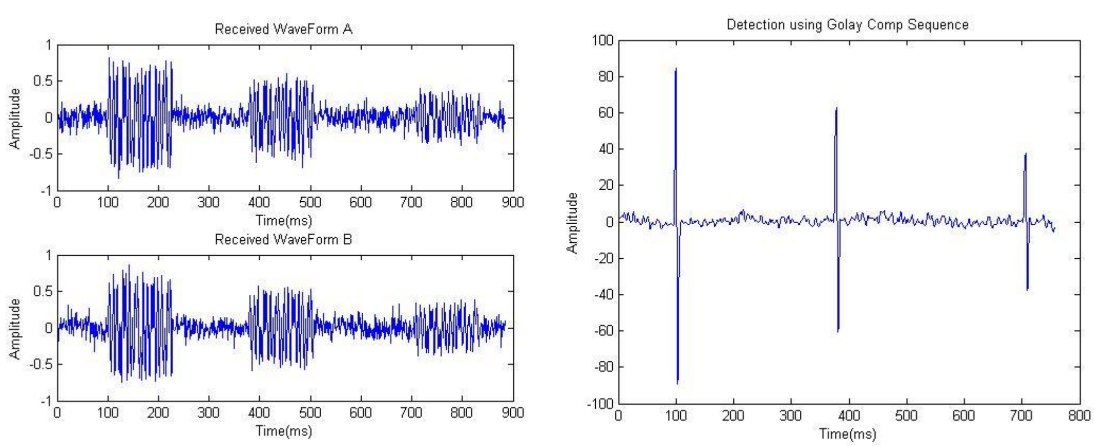


Figure 11*: Length N = 4. Noise comparable to previous runs*

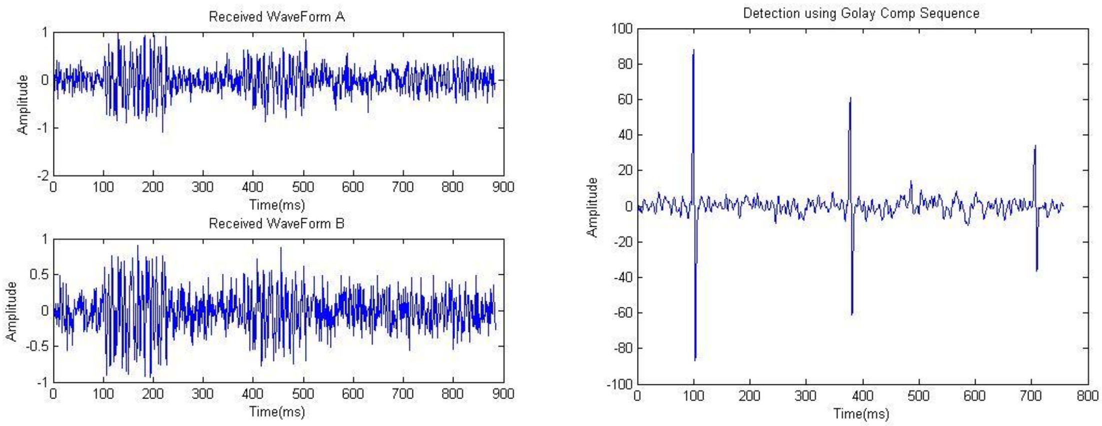


Figure 12*: Length N = 4. Noise variance twice previous runs*

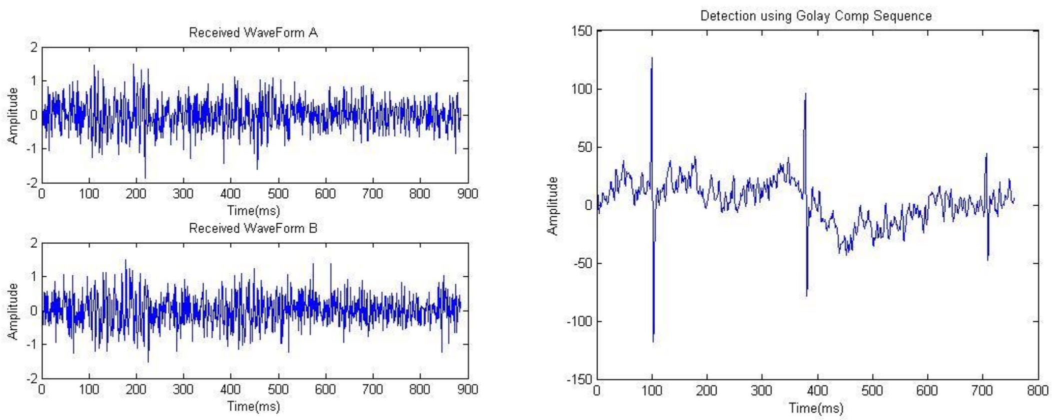


Figure 13*: Length N = 4. Noise variance twice the previous runs*

As we observe, as the noise level increases significantly, the ability to cleanly identify peaks might be compromised. Golay complementary sequences can overcome this difficulty by increasing the length of the Golay complementary sequence. As discussed earlier, the amplitude of the peaks depends on the length of the sequence. Longer the length of the Golay sequence, higher will be the amplitude of the peaks. It is important to note that the simulation has been performed using random Normal generator as the noise source. This might bring small changes to the results between runs.

In the following figures, we try to increase the length of the Golay complementary sequence. The experiment is repeated with N=8.

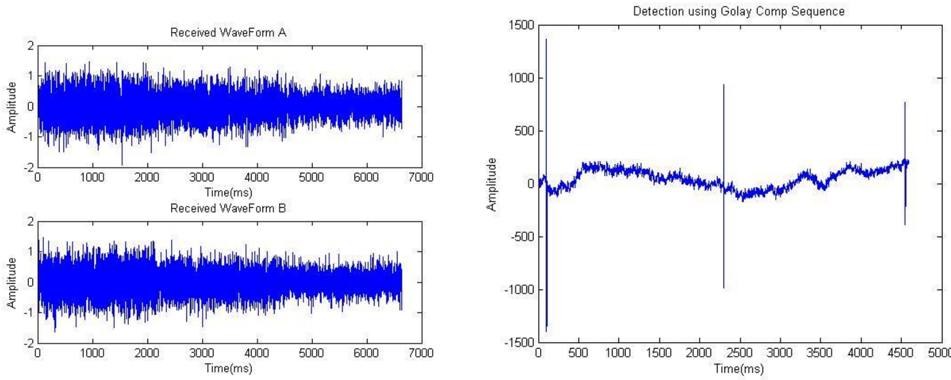


***Figure 14**: Length N = 8. Noise variance twice the previous runs*

As it can be seen from the above figure, if the length of the Golay complementary sequence used is increased, the signal detection capability is highly improved. Even if the signal is buried in noise, it can be recovered. The only disadvantage of using Golay codes of longer length is higher complexity of computation. The resources required to perform the computations can almost be three-four times. This can prove to be expensive on mobile devices as they are not equipped with fast processors. Nevertheless, if required, the process can be optimized. Also the detection cycle is meant to run after a considerable duration of time-period, in the order of fifteen to thirty minutes. The software can also be programmed o use Golay sequences of smaller lengths initially and they iterate the process with Golay sequences of longer lengths of there is no location is detected.

The three methods discussed above have their own merits and demerits. It would be fruitful to use more than one method in an actual implementation. This might be useful in several ways. It can reduce the computational complexity and also the response time of the algorithm. For e.g. If STFT is first used to detect the time points where time-delayed copy of the transmitted signal exists and then Golay complementary sequence is applied only to those parts of the signal, the response can be improved.

The performance of the algorithm can be improved by tweaking one or more of the parameters like Golay sequence length, STFT window size etc. Different environments might justify the use of different techniques and it might be difficult to find a universal solution for all cases which is optimal in every situation. Cost of detection will increase will increase in noise levels.

The length of the Golay sequence might also be instrumental in trying to detect echoes in a large rom. Sound amplitude can be attributed to have a decay which is inversely proportional to the square of the distance. Also the walls and other reflecting objects absorb a large power from the incident sound wave. Thus the reflected wave might be of very small amplitude. Only long Golay sequences can retrieve such signals.

V. Simulation of Techniques to Detect Echoes in Room

We try to use the above discussed techniques in this section to simulate a room environment and detect four major echoes reflected off the four walls of the room. In a real scenario, there would be more reflection coming from more than four surfaces, but for ease of simulation and to provide a proof of concept, the simple case of four walls should suffice.

A sine wave of 2 kHz is spread using Golay codes. Sampling rate of 14 kHz is used for playback as well as recording of the audio signals. The Matlab program allows the user to input the dimensions of the room and the location of the sound source/person carrying the mobile device. There is also an option to simulate the movement of the individual in a bounded region. The program can iteratively simulate the reflection pattern when the sound source is stationed at any point in the bounded region. Random Gaussian noise is added to the waveform.

The example below demonstrates recovery of signal in a room of dimension 10m x 10m. The sound source is located at point (3, 1). The left hand bottom-corner of the room is assumed to be the origin for all measurements. Walls are assumed to absorb 20 percent of the incident sound amplitude. Also the amplitude of the sound wave decreases as inversely proportional to the distance. The above assumptions are fair assumptions to derive a relatively good model for the primary reflections from the four walls of the room.

Length N = 8, Golay complementary sequence is used for the experiment. The received waveform for the transmitted sequences A & B are as shown below.

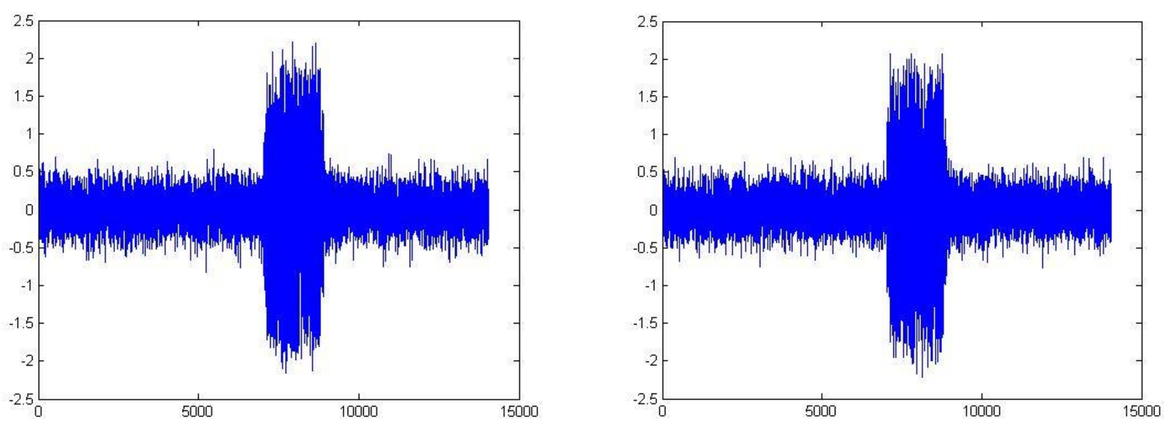


***Figure 15**: Length N = 8. Simulated Received Waveforms for transmitted sequences A & B respectively*

The technique used for detection is exactly the same as described in previously described, namely, time delay estimation using Golay codes. Figure 15 shows the two received waveforms. Interestingly, the echoes are not visible to the naked eye. It simply appears as a huge blob of self-recording of the transmitted pulses.

The reflected pulses are of very small amplitude and are buried in noise. A Golay complementary sequence of longer length will aid in the effective recovery of the time-delayed signal. The presence of the time-delayed signals is established by the presence of peaks in Figure 16. A closer view of the peaks in seen in Figure 17.

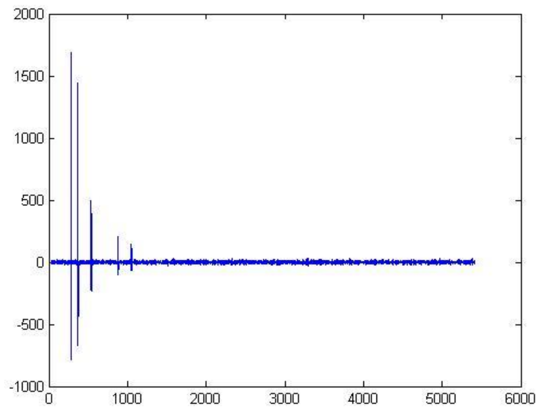


Figure 16: *Length N = 8. TDE using Golay codes*

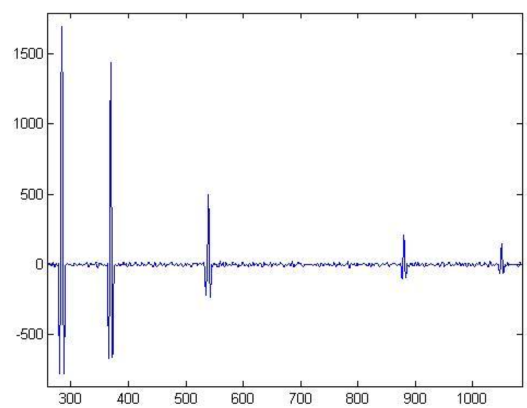


Figure 17: *Closer view of the figure 16*

All peaks comprise of similar pattern of varying amplitude

VI Study of System Response using Actual Audio signals

The above cases deal with ideal scenarios and are not easily replicable in real world scenario using commercial grade speakers and microphones. The cost of instrument quality equipment offsets the feasibility of having this technique on each and every device. The challenge lies in the ability to accurately obtain the same results as in the above simulated case, with a reasonable degree of error to account for the inferior grade low fidelity of audio equipment available on today's mobile devices.

We consider several cases to understand the actual transmitted audio signal when a 1s or 0s are transmitted. Also it is important to study the behavior of the audio system when the transmission sequences comprise of combinations of 1s and 0s. Sudden transitions can drastically affect the behavior of the speaker. It is also better to modulate the code using some sine wave. Speaker characteristics are better for a sine wave than for square pulses.

1. Transmission of series of 1s

   Consider a vector a = [1 1 1 1 1….]. If the vector a is considered as a spreading code and we spread a sine wave using this code, the resulting output would look as shown below, where sin represents an entire cycle.

   TxWave = [*sin sin sin sin sin sin sin* ……]

   Consider the following case, f = 2 kHz and Sampling Rate = 14 kHz. The corresponding sine wave would look similar to the *sin = 2000*pi*f*[0:1/Sampling/0.0005].* The synthetically generated waveform and the actual transmission look as show in the Figure 18.

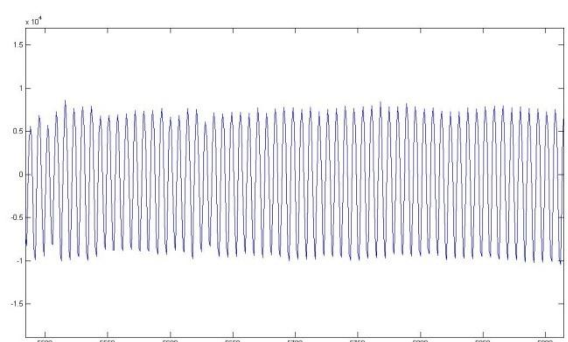


***Figure 18**: Sequence of 1s*

2. Transmission of set of 1s followed by 0s

   In this case, following the previous example if a is represented as a = [1 1 1 1 0 0 0 0]. Then the corresponding TxWave would be represented as follows,

TxWave = [*sin sin sin sin –sin –sin –sin –sin*]

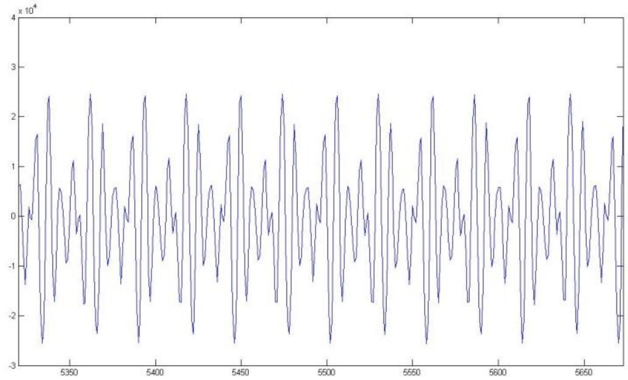


Figure 19*: Sequence of 1s followed by -1s*

3. Transmission of random combination of 1s and -1s

   Now if a is considered to contain the following sequence, a = [1 1 1 -1 1 -1 1 1]. Then the corresponding TxWave be as follows,

TxWave = [*sin sin sin –sin sin –sin sin sin*];

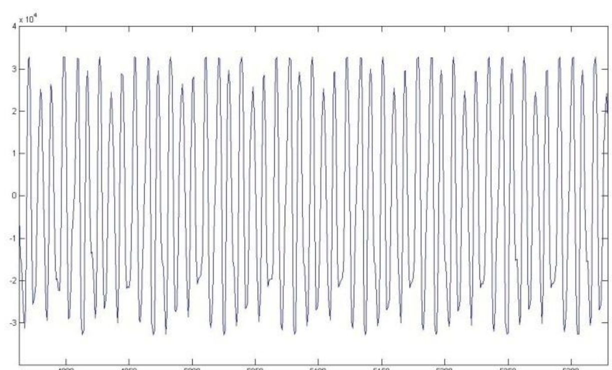


Figure 20*: Sequence of 1s followed by -1s*

VII Implementation of Golay Complementary Sequence Technique

In this section we try to transmit Golay sequence A & B and record the transmitted sequences and analyze them for echoes. We analyze a sample case to present the intricacies involved in the analyses of the waves and extraction of the echoes.

The transmitted sequences look as shown in Figure 21. The Golay complementary sequence is of length N=10, which implies sequence of length 1024 (2^10) sequences.

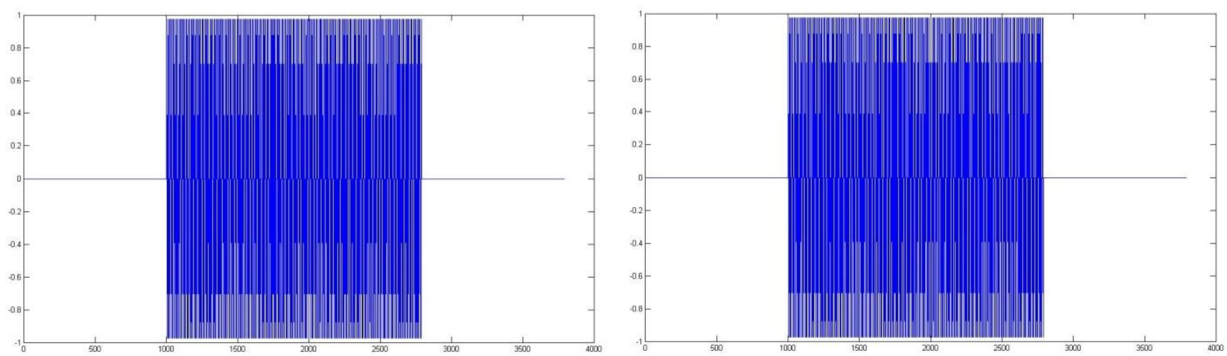


***Figure 21**: Golay A & B Transmission sequences*

The received patterns for the sequences are as shown in Figure 22.

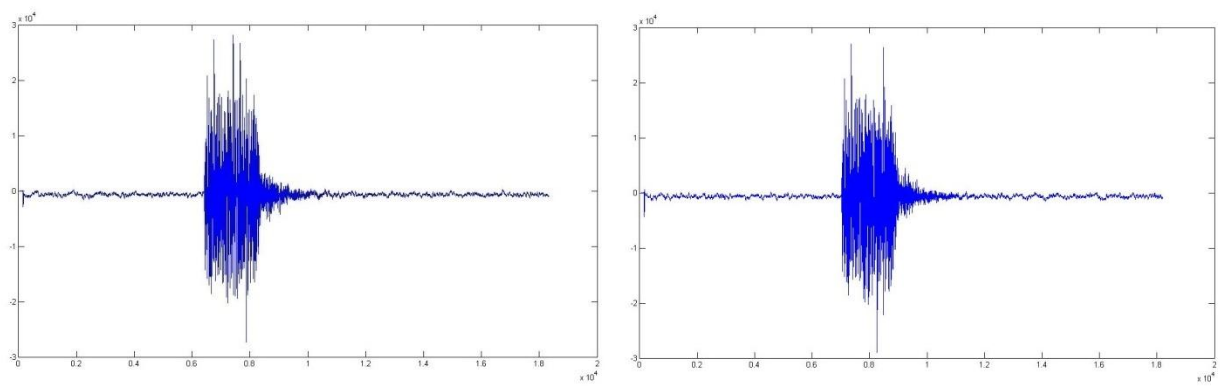


***Figure 22**: Golay A & B Received sequences*

As can be seen from the figures above, the wave pattern is not the same as the transmitted wave. Thus it becomes difficult to use the same techniques as used earlier when the wave patterns are ideal in nature. This problem was also discussed in section VI while transmitting a random combination of 1s and 0s. All of the methods discussed earlier that are used to detect the time-delayed versions of the transmitted signal rely on detecting the time-delayed and attenuated and/or noise added signal to detect the presence of echoes. Thus we have to apply certain normalization techniques and to make sure the received waves are decoded correctly.

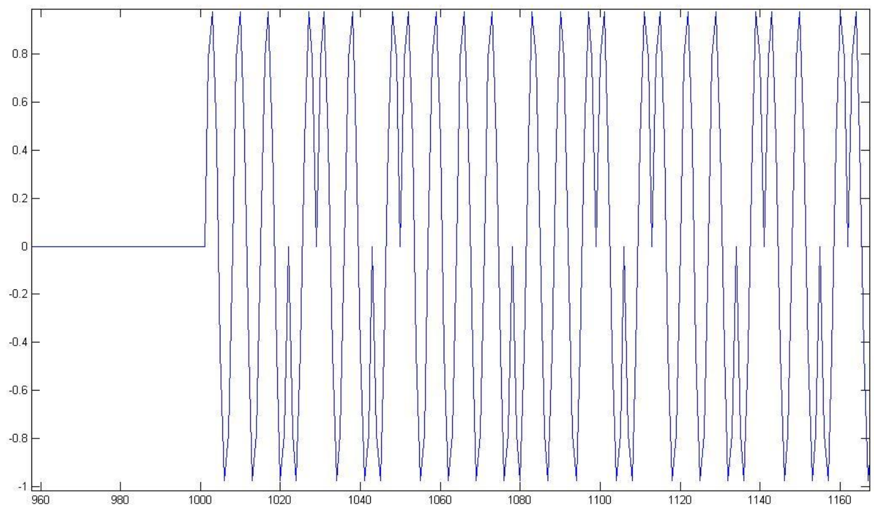


***Figure 23**: First few bits of the Transmitted Sequence*

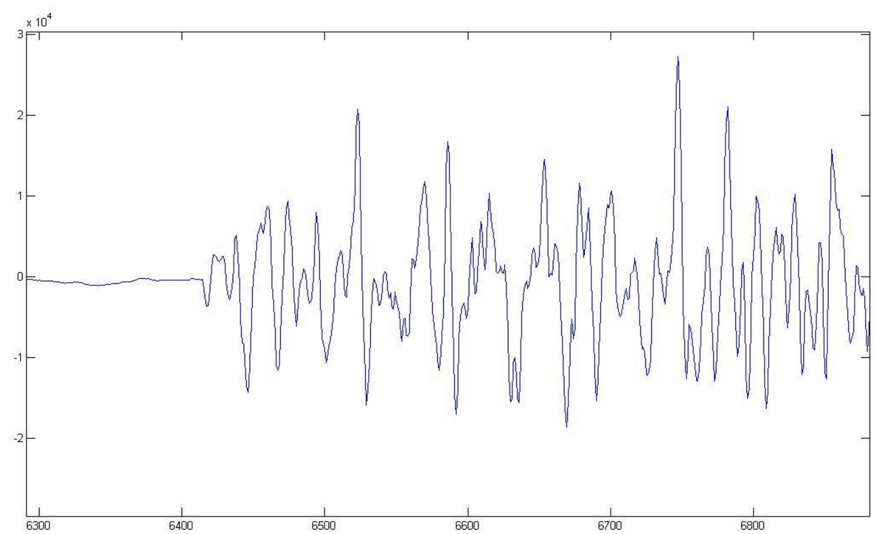


***Figure 24**: First few bits of the Received Sequence*

Note the Visible change in shape of the wave

By applying the steps discussed earlier, we obtain results as shown in Figure 25

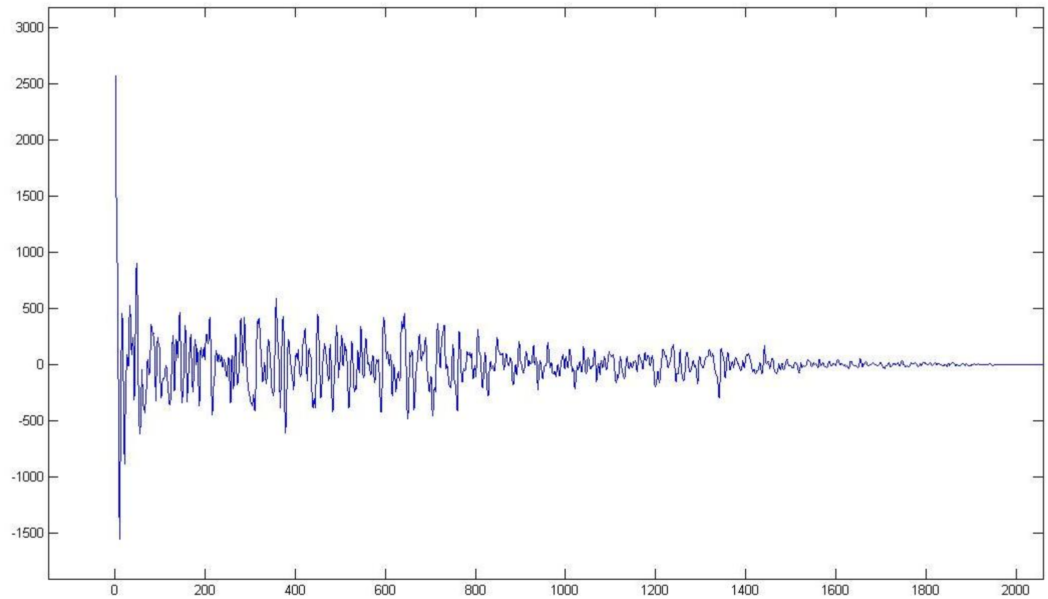


***Figure 25**: Peaks after performing calculations*

The first peak represents the peak due to self recording of the transmitted signals. All subsequent peaks, especially the dominant ones are as a results of the presence of the time delayed copies of the transmitted signal.

VIII Classification for Detection of location

The classification section is relatively intuitive as are most other classification algorithm. We also follow the basic process of having a learning phase where we train the device to understand a particular set of vectors and create a “Feature Vector”. The feature vector can be ideally presumed to be global representation of all possible feature vectors that the device encountered in the learning phase. The detection methods discussed earlier rely on establishing the presence of time-delayed signals by finding peaks in the impulse response of the system. When numerous runs are conducted at a given location, it accounts for the variations in the system like noise etc.

A *Feature Vector* indicating the location of the peaks is created for each location that we intend to recognize or classify in the future. During detection phase, the results are compared against the Feature Vector database stored on the mobile device. Techniques like Nearest Neighbor classifier can be used to classify the results of the detection phase with the database. A common question that could arise from the Audio Fingerprinting is that of identical rooms having the same Feature Vector. This is an inevitable case but it is unlikely that a person would have his or her office room and conference room identical in all aspects. Also if we incorporate additional measures like storing last known GPS/Wi-Fi coordinates along with the feature vector then we distinguish identical rooms in different places, though the ability to differentiate in the same location would be limited.

IX Current Limitations and Scope for Improvement

The above method would with precision if the speakers in the mobile devices were of instrument quality and all the time-delayed copies of the transmitted signal were recorded accurately by the microphones. Such precision is currently available in today's mobile phones. Also, it would be helpful to use ultrasonic or infrasonic pulses for transmission so that they do not interfere with the routine life of the user.

The ability to reproduce the transmitted waves would bridge the gap between synthetic simulation and practical simulation performed. The synthetic simulation performs efficiently even in the presence of noise and overlap of reflected signals. Thus it is not impractical to assume that synthetic results are reproducible on the mobile devices. Advancement in technology and falling costs of high fidelity commercial grade might allow for this.

A large number of simultaneous echoes which are arriving at the same time might introduce diversity problems and the received wave might be rendered useless. But we can assume that the dominant echoes can be easily detected and multipath reflections with more than two reflections would be adequately separated from the primary dominant echoes. In the trial runs that were conducted, it was observed that the primary echoes are distinct and can be detected without any advanced signal processing techniques. By using Golay complementary sequences we can do a much better job than by simply detecting a short pulse and listening for time delayed samples of the pulse. The primary one or two pulses might be detectable beyond which the received signal would look clobbered and would call for the need for more advanced technique like Golay sequences.

IX Android Framework & Implementation

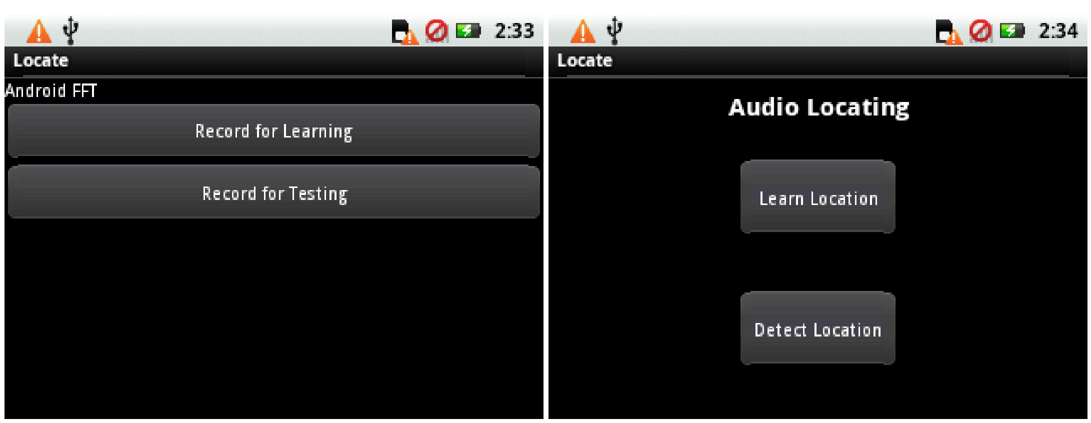
For Proof of concept implementation, the Audio Fingerprinting was implemented as an Application the Android Platform. Motorola supplied us with a new phone which was equipped with a 1 GHz processor, capable of performing the calculations needed for learning and detection phase at commendable speeds. The whole program was implemented using Java. The learning phase included repeated transmission of pulses and analyzing the received signal for time-delayed version of the transmitted pulse. The application performs all the steps as discussed in this report.

The application was created in collaboration with personnel from the Motorola Mobility Team, who provided vital inputs to help improve the speed and performance of the application. Modern techniques provided by the Android Platform namely AsyncTask were used to program the application. The application was multithreaded with the UI coded on the main thread and the transmission and analysis of the waves on different threads that were spawned as required.

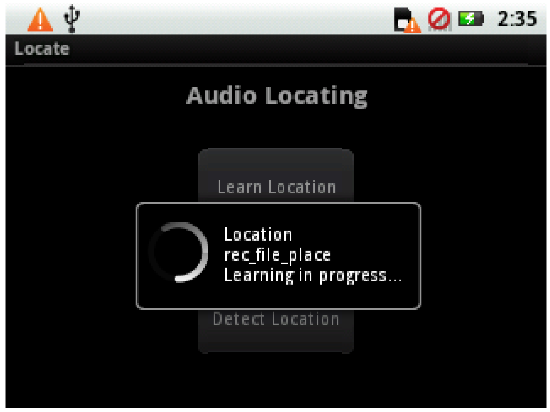
Screenshots of the application are provided to familiarize the reader with the application. The UI was simple in nature as it would be never be deployed as an actual "App" on the device but would rather run as an Android background service and run whenever required and perform the necessary tasks.

The various steps of the application are demonstrated in the figures shown below

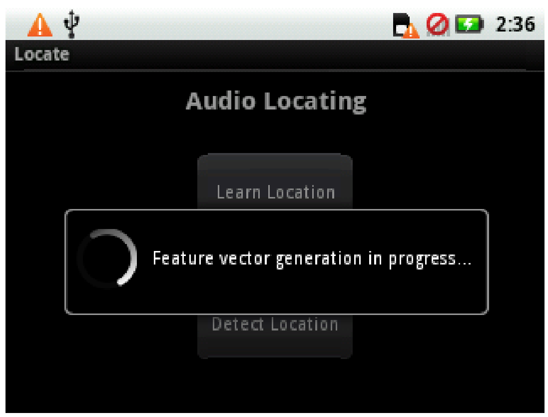
1. **Application Landing Page (V1 and V2 of the application)**



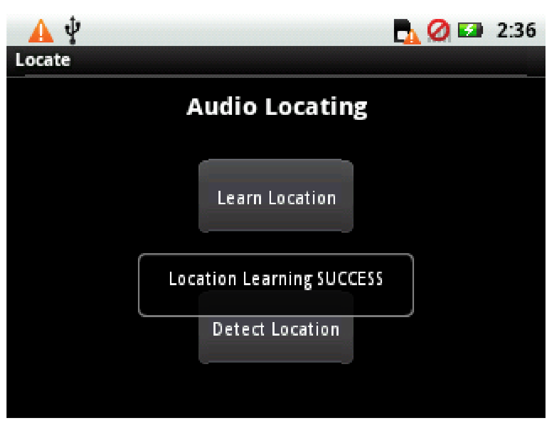
2. **Transmission of learning Pulses**



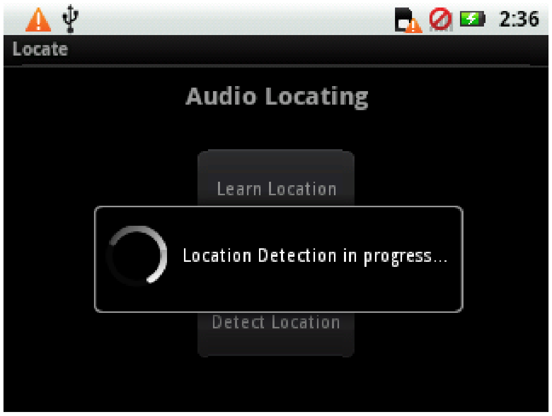
3. **Feature Vector Generation**



4. **Completion of Learning Phase**

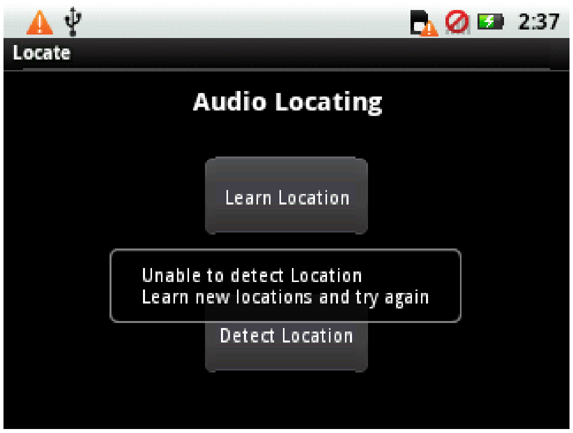


5. Detection Phase



6. Error Handling

If a location is detected it is programmed to perform a set of Tasks in the background. In case the device is unable to classify the received waveform uniquely to a location or has no feature vector match, its discards the run and displays a message as shown below



X Future Work

The accuracy of prediction can be greatly improved by incorporating higher quality of audio system on the mobile devices. Also relying on other location pointers like last known GPS location or Wi-Fi location we can eliminate false matches to any particular location due to similarity in the rooms where the measurements are close to each other for audio.

Many other research projects have also focused on studying the ambient noise and building a classifier that would rely on the noise levels at a particular location. Depending solely on noise levels might be unreliable as noise levels can vary by a large number during the day. There are other techniques where rooms are equipped with sound sources and the mobile devices listen to the audio transmission from these sound sources. But such a technique would call for audio infrastructure to be set up in a room. Our method does not rely on any particular infrastructure requirements. Thus it can be easily deployed anywhere and in almost any phone with a reasonable processing capability.

The idea is extremely promising and can allow the mobile device to act in an extremely intelligent way without any intervention from the user. The user has to simply program the phone and assign tasks for given location. Once the phone detects that it's in a particular location, it can run the tasks that are meant for that location.

References

1. Edgar Berdahl, Julius Smith,*"Impulse Response measurement Using Golay Complementary Sequences"*, www.cnx.org
2. Fan Di, Cao Maoyong, Sun Nongliang, *"Time Delay Estimation Based on Weiner Filter in Ultrasonic Detection of Sediments in Drilling Hole"*, 2009 Second International Workshop on Computer Science & Engineering
3. Atri Manda et.al, *"Beep: 3D Indoor Positioning Using Audible Sound"*
4. Esko Olavi Dijk, *"Indoor Ultrasonic Position Estimation Using a Single Base Station"*, Thesis
5. Short Time Fourier Transform, Wikipedia
6. Java implementation of FFT, www.cs.princeton.edu