Real-Time Sound Source Localization

Project Report for EE 586

Team Members
Jin Fu Liou
Kunal Joshi
Gaurang Vador
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Abstract</td>
<td>03</td>
</tr>
<tr>
<td>2. Introduction</td>
<td>04</td>
</tr>
<tr>
<td>3. System Description</td>
<td>06</td>
</tr>
<tr>
<td>3.1 Microphone</td>
<td>07</td>
</tr>
<tr>
<td>4. Theory and Selection of Algorithm</td>
<td>09</td>
</tr>
<tr>
<td>4.1 Cross Correlation(xcorr)</td>
<td>10</td>
</tr>
<tr>
<td>4.2 Time Domain Analysis –A Wrong Approach!</td>
<td>10</td>
</tr>
<tr>
<td>4.3 Generalized Cross Correlation (GCC)</td>
<td>13</td>
</tr>
<tr>
<td>5. Implementation and Results</td>
<td>17</td>
</tr>
<tr>
<td>5.1 Advantage of using GCC PHAT with FFT instead of Time-Domain Correlation.</td>
<td>18</td>
</tr>
<tr>
<td>5.2 Some thoughts on Low SNR Problem and SNR Threshold for Time Delay Estimation</td>
<td>19</td>
</tr>
<tr>
<td>5.3 Selection of Sampling Frequency</td>
<td>20</td>
</tr>
<tr>
<td>5.4 Quantitative Results</td>
<td>21</td>
</tr>
<tr>
<td>5.5 Clustering</td>
<td>22</td>
</tr>
<tr>
<td>5.6 Intuitive Visual Display</td>
<td>22</td>
</tr>
<tr>
<td>6. Conclusion</td>
<td>23</td>
</tr>
<tr>
<td>7. Future Work</td>
<td>24</td>
</tr>
<tr>
<td>8. References</td>
<td>25</td>
</tr>
</tbody>
</table>
1. Abstract

Accurate localization of multiple sound sources in noisy and reverberant environments is indispensable for any signal processing application and microphone array-based high quality sound capture. For localizing sound sources, CSP (Cross-power Spectrum Phase analysis) / GCC PHAT (Generalized Cross Correlation with Phase Transform) method has been proposed. The CSP analysis method has been demonstrated to perform well even in moderately adverse conditions. The CSP method localizes a sound source as an intersection point of sound directions estimated using different microphone pairs. Exploiting the linearity of GCC to accumulate information from plurality of frames in time domain, a method has been implemented that can localize multiple moving sound sources in the same condition using three microphones. Real time localization and an intuitive display of the localized sound source have been implemented as a part of this project.
2. Introduction

Time Delay Estimation between replicas of signals is intrinsic to many signal processing applications. Depending on the type of signals acquired ranging from human hearing to radars various Time Delay Estimation methods have been described in literature. Sound source localization (SSL) systems estimate the location of audio sources based on signals received by an array of microphones. With proper microphone geometry SSL systems can also provide 3D location information and can easily recognize/isolate sound generating sources such as speakers, TV, conversations etc. The other example of SSL is to locate sound sources so that a robot can interact with detected objects. Rotating a microphone in a conference room to isolate and process a particular speaker is another example where SSL systems can be implemented.

The project achieves real-time localization of multiple sound sources. Real time implementation involved making trade-offs between size of frame and accuracy, reducing the number of microphones to increase the speed of the system, clustering data from multiple frames to accurately localize multiple sources. The details of these trade-offs have been documented in this report.

The direction of the sound source can be obtained by estimating the relative Time Delay Of Arrival (TDOA) between two microphones. Peak levels for each microphone signal are analyzed, from which a time delay between signals can be found. The location of the source relative to the
microphone array is calculated using this delay, and this location is displayed on the computer screen. The procedure for localization of multiple sound sources by the TDOA method is:

1. Estimation of delays of arrivals (Estimation of directions of multiple sound sources).
2. Localization of multiple sound sources by clustering delay of arrivals.
3. Display the sound location in real time.

In this project, various time delay estimation methods were exhaustively tested using Matlab and the best method was implemented on the TI DSP C6713 board with an add-on audio daughter board. The methods are time domain cross-correlation (CC), general cross-correlation (Unfiltered), general cross-correlation with phase transform (GCC-PHAT), smoothed Coherence Transform (SCOT)[3], modified SCOT filter of Cross Power Spectrum (CPS-m), maximum likelihood estimator (ML), and Roth autocorrelation weighting function (Roth)[4]. Their simulation results are compared when they are implemented on the real samples from DSP board. This report describes the algorithm that was implemented on the DSP board. Finally, the results obtained from a real system are presented to show the effectiveness and practicality of the algorithm.
3. System Description.

Shown below is the block diagram of the Sound Source Localization system implemented. The functions of each block are as follows:

1. Microphone array: Sound is detected by three electret microphones spaced 30 cm apart, as in Figure 3. Each microphone is powered by a 4 V power supply and the acrylic board on which the microphones are placed is covered by an acoustic foam sheet to reduce the noise due to reflection.

2. Amplification: Each microphone signal is amplified using pre-amplifier boards available in the lab to an adequate level. The amplifier also removes the DC offset voltage from the signal.

3. A/D conversion: Analog signals are converted to digital by a Texas Instruments ADK-6713 board attached to the PC at a sampling rate of 96 kHz.

4. Signal Processing: Incoming data is processed using CC-Studio to calculate time delay and source location and location estimate for each frame is sent to PC via RTDX.

5. Output Display: The LabView program provides a front panel to display the results of positions both from clustering algorithm and DSP, and a background display panel to show the live feed from webcam.
3.1 Microphone

The design of this system involved selecting the proper microphone element. The first concern was the directivity, indicating how sensitive it is to sounds arriving at different angles about its central axis. Since the SSL system needs an equal response for sounds from any direction, omnidirectional microphone was selected. A few other references have suggested electrets
microphones available at radio shack which give a flat response over the frequency range of 20Hz to 4 KHz (human voice audio range) as shown in figure 4. Although some dynamic omnidirectional microphones were available in the lab, the frequency response, as shown in figure 4, was not as flat as the electret microphones as shown in figure 5. Since the electret microphones have the advantage of good frequency response range, small form factor and also since the microphones have already been implemented in SSL system by others, they were used in the project.

Figure 4 Frequency response of Radio shack 33-3030

Figure 5 Frequency response of Radio shack 270-092[1]
4. Theory and Algorithm Selection

The direction of a sound source can be obtained by estimating the time delay of arrival (TDOA) between two microphones. Figure 6 illustrate the specification of TDOA estimation.

For an original signal $s(t)$ from a source $S$ impinges on microphones 0 and 1. $n(t)$ being the noise source and $h(t)$ being the impulse response.

$$x_i(t) = h_i(t) \times s(t) + n_i(t), \quad i = 0, 1$$

The easiest approach to estimate the time delay is to choose the index while performing time domain cross-correlation which gives maximum value, and to convert this index to geometric angle using simple algebra. But this approach is only useful in an ideal setting with no noise and hence the system is not robust and turns out to be totally useless in practical settings.

Another approach is to use the frequency domain to get the phase information of the correlated outputs. The correlation is calculated using inverse Fourier transform of the cross power spectrum. Once a weighing factor is introduced taking into consideration the statistics of the source signal and noise various Generalized Cross Correlation methods can be formed and are found in literature.
Over the course of selecting the right method, the methods that were tested and the one that was finally selected and implemented are described in detail below.

4.1 Cross Correlation (xCorr)

The first approach to estimate the time delay is to compute the cross correlation (xCorr) between the received signals at two microphones. The figure 8 shows the structure of cross correlation. The location (index) of the maximum peak in the output represents the estimated time delay. The xCorr can be modeled by: [4]

\[ CC_{ij} = \sum_n x_i(n)x_j(n-l) \]

\[ \tau = \arg \max(CC_{ij}(k)) \]

![Figure 7 Estimation of TDOA by xCorr][4]

The sound source direction is then derived by

\[ \theta = \cos^{-1}\left(\frac{c\tau}{Fs} \right) \]

Where \( d \) is the distance between two microphones, \( Fs \) is the sampling frequency (96 KHz) and \( c \) is the speed of sound.

4.2 Time Domain Analysis – A Wrong Approach!

Initially the time domain approach was implemented in Matlab. All of the implementation was
done in MATLAB on the recorded data. In trying to implement a sound localization system using this method time delay estimates either remained constant or were erroneous. Distinct angles should give distinct peaks from the recorded data but when the peaks of the cross correlation were observed it was found that they often spanned a few samples, which was the cause of our poor results. After observing the signal themselves a distinct delay was observed, which did not appear in the cross correlation. Changing the angles by moving often returned the same result as before! The fact that the delay didn't change when angle of the sound source was changed meant that the implementation was wrong or the correlation wasn't specific enough!

Having looked at more sources and particularly [2] generalized cross correlation (GCC) was found as a likely candidate. At low SNR, GCC has an advantage of making the peak distinct and clear. However, at high SNR or in no noise environment Cross Correlation and GCC should give similar results.

Below are few results that were obtained from the time domain analysis which clearly show a complete mismatch of the peaks that we got through the GCC PHAT and the cross-correlation. More so no distinct peak is observed while using Cross Correlation but GCC has distinct peaks from 0 to 50 index.

![Comparison between GCC-PHAT and Cross-Correlation](image)

*Figure 8 a) Comparison between GCC-PHAT and Cross-Correlation for the source at 30 degrees horizontally*
Figure 8b) Comparison between GCC-PHAT and Cross-Correlation for the source at 30 degrees vertically

Figure 8 c) Comparison between GCC-PHAT and Cross-Correlation for the source at 60 degrees horizontally

Figure 8 d) Comparison between GCC-PHAT and Cross-Correlation for the source at 120 degrees vertically
Hence, as can be seen from the above results, it was difficult to get the right estimate of the peak in the cross-correlation in order to get the correct delay since there was never a distinct peak. As a result, the angles were erroneous. GCC was tested, found to be suitable and free of the errors that were in cross correlation and was implemented successfully.

4.3 Generalized Cross-Correlation (GCC)

A way to sharpen the cross correlation peak is to whiten the input signals by using weighting function, which leads to the so-called generalized cross-correlation technique (GCC). The block diagram of a generalized cross-correlation processor is shown in Figure 10. The procedure of GCC has received considerable attention due to its ability to avoid spreading of the peak of the correlation function [3, 4, and 6].

\[
x_i(t) = h_i(t) * s(t) + n_i(t), \quad i = 0, 1
\]

\[
G(f) = X_0(f)X_1(f)^*
\]

Where \(G(f)\) is the frequency domain cross correlation of the two signals.

However, to reduce the noise and reverberation in the environment, a normalizing factor is applied, that preserves the phase information of the cross correlation.
\[ G_{PH}(f) = \frac{X_0(f)X_1(f)^*}{\|X_0(f)\| \|X_1(f)\|} \]

Where \( 1 / \|X_0(f)\| \|X_1(f)\| \) is the weighing function.

Other weighing functions like the Roth autocorrelation (Roth) weighting function are used sometimes:

\[ \varphi(t) = \frac{1}{S_i(t) \times \text{conj}(S_j(t))} \]

And the Smoothed Coherence Transform (SCOT) filter [3]

\[ \varphi(t) = \frac{1}{\sqrt{S_i(t) \cdot \text{conj}(S_i(t)) \cdot S_j(t) \cdot \text{conj}(S_j(t))}} \]

The Matlab GCC function provides the computation of a pre-whitening filter onto the cross-power spectrum in order to weigh the magnitude value against its SNR. The weighted Cross Power Spectrum (CSP) is used to obtain the cross-correlation in the time domain with an inverse Fourier transformation. The result is normalized between \([-1 \quad 1]\). The following **figure 10** shows the simulation result of MATLAB with real time sample recorded from board at 48 kHz for the fixed position sound source. Different filter functions- unfiltered general cross-correlation (Unfiltered), phase transform (PHAT), smoothed Coherence Transform (SCOT), modified SCOT filter of CPS (CPS-m), maximum likelihood estimator (ML), and Roth autocorrelation weighting function (Roth)-are compared. The PHAT, SCOT and CSP-m show sharper peaks at the magnitude value of CSP. Besides, the reference [2,3,5,6,7] say the PHAT method has better performance in simulated Gaussian distribution noise and actual noisy environments. The PHAT function is also less computationally intensive and hence very useful for real time system. Therefore, the PHAT function was adopted in the project.
Figure 10 Simulation result of Matlab GCC function

Figure 11 Simulation result of xCorr and GCC-PHAT

Figure 11 shows the simulation result of xCorr and PHAT with real time sample recorded from
board at 48 kHz for the fixed position sound source. The PHAT method sharpens the peaks at the correct time delay compared to the xCorr result in actual real time signal.
5. Implementation and Results

Having looked into the principal of the implemented Generalized Cross Correlation algorithm, the pseudo code is detailed below.

Pseudo code / Implemented Algorithm:

1. Fill 8192 size buffer at a rate of 96 KHz with preamplified sound signal from 3 different microphones, forming 2 pairs needed for vertical and horizontal localization.

2. Optimized FFT function provided by TI is used to calculate the FFT of the 3 buffers.

3. FFT of Left buffer is multiplied with conj (FFT of Right) buffer, and similarly FFT of Right buffer is multiplied with conj (FFT of Bottom) buffer.

4. Then these two products are divided by the weighing function.

5. This leaves only phase information in the buffers.

6. An IFFT of these 2 buffers is performed using the same optimized FFT.

7. The output then gives a buffer with a peak at the index which corresponds to the time delay between the two duplicate signals.

8. Each delay corresponds to a particular angle. In order to optimize the code a lookup was created using the relation

   \[ \theta = \cos^{-1}\left(\frac{cT}{Fs} \right) \]

   There is a one to one mapping between the delays and the angles. Hence the lookup table is implemented to reduce computation time for every frame.

   In order to display the values in real time on the screen, we used LABVIEW. The angle values were transferred to the computer. Data transfer between the DSP and the computer is handled by TI’s Real Time Data Transfer (RTDX).
5.1 Advantage of using GCC PHAT with FFT instead of Time-Domain Correlation.

Since the PHAT function will be used to get the TDOA, \( \tau \), by finding the maximum value of the cross power spectrum, the CSP coefficients are derived from the following equation:

\[
csp_{\mu_i}(k) = \text{DFT}^{-1} \left[ \frac{\text{DFT}[s_l(n)]\text{DFT}[s_l(n)^*]}{||\text{DFT}[s_l(n)]||||\text{DFT}[s_l(n)]||} \right]
\]

\[
\tau = \arg\max(csp_{\mu_i}(k))
\]

This equation will be realized by two DFT, one inverse DFT and absolute calculation. There is also the other way to realize this CSP function:

\[
csp_{\mu_i}(k) = \text{DFT}^{-1} \left[ \frac{\text{DFT}[\text{cor}(s_l(n)s_l(n))]}{||\text{DFT}[s_l(n)]||||\text{DFT}[s_l(n)]||} \right]
\]

This equation will be realized by one DFT, one cross correlation, one inverse DFT and absolute calculation. The speed of computation can be compared by MATLAB “tic” and “toc” function. The simulation results of the first methods with 19 frames (1 frame has 8192 samples) is 0.356874 seconds and the necessary time for the second method is 0.139220 seconds.

![Figure 12 Simulation result of 2 realization of GCC-PHAT](image-url)
Figure 12 shows the simulation result of these two realization method. Both these two method can get the same result of max CSP index. Since the cross correlation equation will expand the original frame size into twice, the max CSP value will be right at the middle for the first method. On the other hand, for the second method, the max CSP index will locate at both most left-hand and right-hand side. Both these two side will be affected by the rectangular window effect. However, for realization in the real time system, we still choose the second way, but ignore the most 2 samples at the most left-hand and right-hand sides.

5.2 Some thoughts on Low SNR Problem and SNR Threshold for Time Delay Estimation

Generally, room reverberation is considered as the main problem for TDE. Moreover, acoustic background noise may further decrease the performance of time-delay estimators. The performance of TDE is always been affected by the reverberations in a room. The problem becomes more challenging once room reverberations rise. In a highly reverberant room, all the known TDE methods become unreliable and even fail. Few early studies have investigated the TDE problem in the presence of a few correlated additive echoes. In particular, the quantitative behavior of the estimator variance for reverberation can be explained naturally in terms of an equivalent signal-to-noise ratio (SNR), which treats the reverberant energy at the microphone output as undesirable noise. Namely, the high level of reverberation causes the low value of SNR.

Finite time measurement causes the estimated cross-power spectrum incurs a certain variance, which may affect the accuracy of TDE. It can lead to a large error when the actual observation interval is short. In many cases of practical interest, however, the assumption of long observation interval is inconsistent with other prevailing conditions, such as assumptions of stationary processes and constant delay will only be satisfied over a limited time interval. There is actually a tradeoff between observation time and SNR in the TDE problem. In the case where SNR is low, the long observation time is required to ensure that the accuracy can be obtained. On contrary, the higher the SNR, the shorter observation interval is needed. The observation interval decided was of frame size
to be 8192 samples, sampled at a rate of 96 KHz. The environment where we ran and simulated our project is our lab, which is an environment where we are able to achieve a very high SNR. This encourages us to not choose a very large frame size which is very favorable for optimizing our code as the FFT will be performed over a comparatively smaller frame size. In order to decide an optimum frame size we took the recorded data, sampled at 48Khz at various angles and tested it on MATLAB. We processed it for different frame sizes starting from 1024 to 8192samples per frame. It gave us very good results for a frame size of 8192, and even had a fairly fast processing in MATLAB, which we tested through the “tic” and “toc” functions in MATLAB. However, in order to increase our resolution and have a lot more accuracy of angles we decided to sample the incoming data at a rate of 96 KHz.

5.3 Selection of Sampling Frequency

The selected sampling rate of 96 KHz gave higher resolution of angles as well as a good visual display. Initially the sampling rate selected was 48 kHz which gave good resolution of 5 degree but with 96 KHz the resolution increased to 1 degree. The sampling rate plays an important role in the range of angles that we can get for the Horizontal and Vertical angles that we are trying to calculate. Our system comprises of 3 microphones such that the adjacent microphones are 0.30 meters apart. These two factors, i.e., the distance between the microphones and the sampling rate of the DSP play a pivotal role in the resolution that we may get.

For 48 KHz, we are able to achieve 85 different angles for both horizontal and vertical angles, that span the entire range of 0 degrees to 180 degrees, in both dimensions. This can be calculated as follows:

\[
\text{The maximum delay between 2 adjacent microphones} = \left(\frac{F_s \times d}{c}\right)
\]

Where

\(F_s = \text{Sampling Frequency} = 48 \text{ KHz}\)

\(d = 0.30 \text{ meters}\)
\[ c = \text{Speed of sound} = 340.29 \text{ m/sec} \]

Thus the maximum delay that we get between two microphones is approximately 42 from one side. Hence the delay between the two microphones will range from -42 to +42 while spanning 0 degrees to 180 degrees. This results in 85 different delay measurements over the entire range. And each delay corresponds to different angles. Thus, each subsequent delay results in the difference of the angles by 4 to 5 degrees. We wanted to get this value to approximately 1 degree per delay. Hence we doubled the sampling rate to 96 KHz, and had a drastic increase in our resolution. With other parameters remaining constant (i.e. the speed of sound and the distance between microphones), the sampling rate of 96 KHz resulted in giving us approximately 170 different delays where each delay corresponds to a different angle over the entire range of 180 degrees. Thus, each subsequent delay results in the difference of the angles by approximately 1 degree.

### 5.4 Quantitative Results.

The table below shows the actual angels vs. measured angles.

<table>
<thead>
<tr>
<th>Actual Angle</th>
<th>Measured Angle</th>
<th>Error (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>83</td>
<td>3</td>
</tr>
<tr>
<td>90</td>
<td>91</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>112</td>
<td>2</td>
</tr>
</tbody>
</table>
As can be seen with 48 KHz, angles in the range 110 degrees > source > 80 degrees were more prone to errors and this error was resolved using sampling frequency as 96 KHz.

5.5 Clustering.

A clustering algorithm using histogram has been implemented to remove noisy data samples. The pseudo code for the same is described below.

1. A histogram of past 10 frames is calculated every 5 frames.

2. The max value of angle is calculated from the histogram with angle within a radius of 2 degree of each side.

3. More weight is given to the current angle.

These clustered angles for both vertical and horizontal were sent using RTDX using 2 more channels but the speed of data transfer reduced drastically and hence the clustering algorithm had to be implemented in LabView.

5.6 Intuitive Visual Display

As shown on the first page a webcam was used as a live video feed and the angle data from the TI DSP were plotted on the source of sound displayed on the video. To achieve this, the graph displaying the angular data was scaled and mapped to the webcam field of view. Having less resolution within this viewing angle was one of the main reason to switch the 96 KHz. switching to 96 KHz gave higher resolution within this viewing angle and gave a better visual display.
6. Conclusion

The GCC-SCOT, GCC-ROTH and GCC-PHAT and the Maximum Likelihood Estimator are useful \textit{ad hoc} techniques for analyzing time delay characteristics between two random processes. Proper interpretation of the SCOT requires knowledge of the model of the received processes. All these approaches account for analysis in the frequency domain. From our attempts in order to get the delay between the two signals it was realized that analysis in time domain is not one of the best solutions. It just doesn’t work well for real time applications where there is noise. Hence time domain analysis involving cross-correlation and least square must not be carried out in order to get the right estimate of delay between the two signals. However, at high SNR, or at no noise cross correlation and GCC should be similar.

The TDE methods like the SCOT, Cross Correlation, Maximum likelihood etc described in this report work well in case of high SNR. However, in the actual noisy environment, PHAT seems to be the best choice because of its perfect performance in sharpening the correlation peak at the correct time delay and its small SNR threshold. Also, for real time applications the weighting function for PHAT is calculated a lot faster as compared to the weighting functions of the other functions of the GCC family. This reduces our computational complexity. However, when the SNR is below a specified threshold, the performance of all the methods rapidly deteriorates due to large anomalous or ambiguous estimates. While the anomaly and ambiguity effects are fundamental and unavoidable features of the delay estimation problem, independent of the signal processing technique, which make the TDE problem more challenging when the SNR drops under a minimal level. Future work should be focused on the robust time delay estimation with low SNR[2].
7. Future Work:

If some additional features have to be added, the following would be the best candidates

1. Currently only angles in the x-plane and the y-plane (i.e. horizontal and the vertical angles) are being calculated. By adding one more microphone, which the system already has the capability to do so, depth as a dimension can be added to the system.

2. Rotating the camera. A motor underneath the camera would enable it to turn toward the source of sound.

3. Beam forming. Once the sound source is located, sound source isolation and enhancement can be performed.
8. References.


3. A Closed-Form Location Estimator for Use with Room Environment Microphone Arrays - Michael S. Brandstein, Member, IEEE, John E. Adcock, and Harvey F. Silverman, Senior Member, IEEE


5. Use of the Crosspower-Spectrum Phase in Acoustic Event Location - M. Omologo and P. Svaizer.

6. A two-stage algorithm for determining talker location from linear microphone array data - Harvey F. Silverman and Stuart E. Kirtman

