[[1]](#footnote-1)Extract Target Speech Signal from 3 Channel Recordings

Hongyu Zou 7493642

Liu Cheng 7486632

Department of Electrical and Computer Engineering

University of Ottawa, ON, Canada, K1N 6N5

Email: [hzou047@uottawa.ca](mailto:hzou047@uottawa.ca), [lchen156@uottawa.ca](mailto:lchen156@uottawa.ca),

*Abstract*—Currently a good method to extract directional audio is based on microphone array system. A novel technique introduces such three stages for extraction, which involve Time Difference of Arrival calculation, beamformer filtering with directional interference deduction and parametric spectral subtraction for general background noise deduction. In this paper, principles of certain methods will be explored and explained. Furthermore, a specific experiment targets on given piece of audio (captured by three microphone array system and collect three different directional source signals), as a result, audio of source of English female will be extracted. A general look will discuss the quality of result in terms of human perceptual hearing. Related experiment procedure will be given for further development. In addition, the newest Phased Array System Toolbox is also used for taking advantage of modern tool provided by Mathworks company and explained at some key features as well as how to use.

*Index Terms*—Time difference of arrival, beamformer filtering, parametric spectral subtraction, microphone array, multi source audio extraction, phased array toolbox.

# INTRODUCTION

T

his paper targets on application of extracting audio. Mixture audio is captured by three microphone array system. In this stage, the direction of sources will be calculated based Time Difference of Arrival (TDOA) [1]. Through the results of time difference and transform function, angles of direction will be brought to next stage calculation which aims to eliminate certain direction of interference. In our last stage, choosing simple parametric spectral subtraction method to deduct background noise. In following sections, principles are further developed with experimental parameters and results, core code and formulas are illustrated, too. Experiment section is mainly discussing the result of experiment, while part of future work will be explored.

Our stimulate signal source has speeches of English female voice, English male voice and Germany female voice. The distance of each microphone is ; thus we have two spaces in these three microphones. Assuming the source comes from far filed so that calculation base line is not fixed. Moreover, sound speed is assumed as 343 meters per second.

As shown in Figure 1, basic structure of microphone array is assumed in the same line and no consideration of altitude.

2

2

Figure 1 General view of microphone array in application

Phased Array System Toolbox is used for directional interference elimination [2]. This toolbox provides the integrated algorithms and tools for simulation and analysis. Specifically, our work will modify one of official example named as *Acoustic Beamforming Using a Microphone Array* [3]. In this stage, desired speech signal is going through this system with an interference-dominant, noisy environment. Through the pre knowledge of better performance of frost beamformer, this kind frost beamfomer will be used in second procedure.

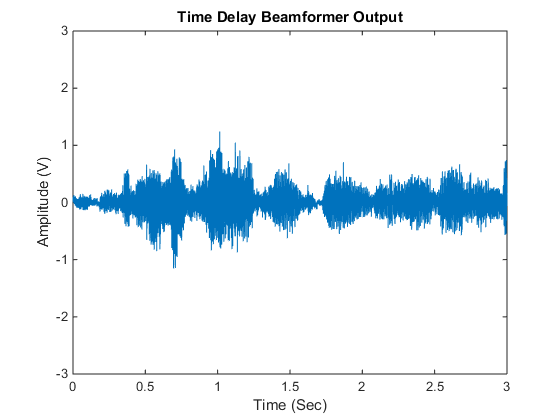


Figure 2 Official Time Delay Beamformer Output

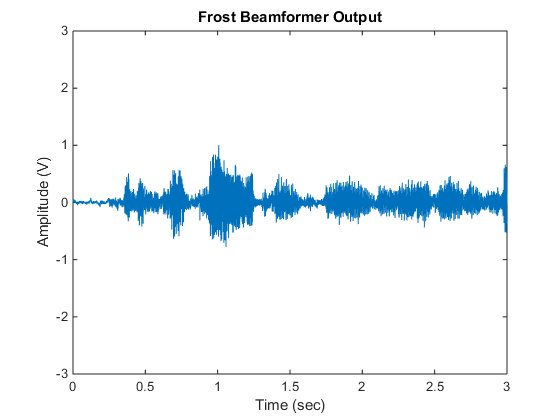


Figure 3 Official Frost Beamformer Output

As shown in Figure 2 and Figure 3, official test illustrates the better performance of frost beamformer for directional interference elimination.

# Basic Background

TDOA algorithm is seen as improvement of TOA (Time of Arrival). TDOA method is originally an effective method to detect location of moving object. Experiment takes use of three microphone with three sources [4]; thus three TDOA should be concluded for each source. The advantage of TDOA includes multiple time delay and synchronize error. Beside TOA timestamp, TDOA only keeps length of time difference. In terms of two dimensional distance expression, it is concluded as [5]

(1)

Microphone is well-known as sound pressure level pick-up which transform to voltage variation as electric signal. Generally, microphones have an Omni-directional polar pattern which means they are not extremely sensitive to certain direction. To locate direction of source, a method is taking use of TDOA technique. Cross-correlation function measures the similarity of signals which have different start point, defined as

(2)

Where denotes the time difference, to calculate maximum value of , will be got. Furthermore, as we preset the space of each microphone by , simple expression for horizontal signal propagation will be

(3)

However, in our application, propagation is from two-dimensional propagation such that formula (3) is modifies as

(4)

Which is an approximately calculation for angle , denoted in Figure 4.

Figure 4 Calculate angel from time difference

The assumption which is used is based on following sets as pre knowledge which reduces the complexity of experiment:

* *all the sources are spatially stationary*
* *free field conditions are simulated, as opposed to reverberant conditions with possibly long echoes, this means that the impulse responses / frequency responses between the sources and the sensors are simple, i.e., they correspond to pure delays*
* *plane wave propagation is assumed, meaning that we consider that the sources are in the far field, with no near propagation field effects*
* *there is no physical object between microphones*
* *we assume a known and constant speed of sound here, but in practice the speed of sound is a variable that depends on temperature and humidity*
* *time-frequency statistics of the noise are stationary, and the simplest type of noise was used: white Gaussian noise. In addition, 2 seconds of noise-only signals are available at the beginning of the files.*
* *offline processing is possible*
* *it is often not possible in physical systems to have such large distances between microphones and large distances can help in the detection of the angles and in the target source extraction*

In terms of noise deduction, there are numbers of choices of potential filters and methods. While in our experiment, after comparing several different methods with respect to Signal to Noise Ratio (SNR), parametric spectral subtraction is easy to realize and it has satisfied consequence, too. After filtering out the English female voice, parametric spectral subtraction keeps acoustic characteristic of voice without much distortion. Therefore, some brief introduction for this kind of method will be referred to in later session with experiment result.

# calculate time difference of arrival for directional filtering

Time differences of arrival (TDOA) is also known as an effective method to evaluate direction of audio source. In our experiment, 25cm microphone spacing is generally a large spacing; thus using generalized cross-correlation with phase transform (GCC-PHAT) is a good choice [6].

Since the whole audio piece is already known, it’s possible to use Short Time Fourier Transform (STFT) to get this job for analyzing. To construct , illustrated in background session, the peaks in indicate the arrival of three sources. Specifically in this project, there are three peaks and indicate for each of them. As a result, three mixture audio will generate nine peaks in terms of different comparison. While it’s possible to get time differences by only two mixture audio, far field effect actually has less impact on different choices.

For STFT, mixture signals can be denoted as , also known as time-frequency bin (the length of each bin is preset as parameter). The source signals will be denoted as , where and are bin indexes. In addition, is steering vector for arrival time of -th source, and denotes background noise in mixture signals. Therefore, mixture signals can be expressed as

(5)

After previous discussion, is thus modified to

(6)

Note here indexes are from to and to for and , respectively. We actually need to provide better performance, in other word, more data as possible as we can. Because of that, taking all frames is the best choice (to iterate all frames along the signal). This part is then described as

(7)

The exist method maps calculation of cross-correlation with time-frequency bin, forming *empirical covariance matrix* , thus

(8)

Then we can apply it to local angular spectrum

(9)

In our experiment, *BSS Locate* toolbox is used for source localization in stereo convolutive audio mixtures by Blandin’s work. As it is already been open source code, it’s easier to do the modification based on related libraries. In fact, not only GCC-PHAT method is involved in, but also such method like GCC-NONLIN. We only focus on GCC-PHAT though. From the input of function, five parameters are specified:

* *x: samples which contain two mixture signals*
* *fs: sampling rate, unit in Hz*
* *d: microphone spacing in meters*
* *nsrc: number of sources*
* *local: angular spectrum function (such as GCC-PHAT, GCC-NONLIN, etc)*

Obviously, the input of signals contain two pieces of audio which forms as two row of vectors. Each vector include 12s audio signal, while sampling rate is 16000 Hz. Thus, totally 19200 samples are used. Spacing is 25 cm which is indicated by project assumption. Finally, number of sources is already known as 3 and spectrum method is GCC-PHAT. In the meanwhile, a dependent library is involved as *stft\_multi* which will be used to measure STFT bin and related operation based on multiple bins. The output of function is not used since the average summation of mixture audio is not needed.

After calculated TDOA, using formula (4) to calculate angles for each source.

# Beamformer Filtering for Eliminating Directional Interference

(cont.)

# Parametric Spectral Subtraction for Background Noise Deduction

# Experiment Result

## General parameters setting

In this part, general parameters are set up for future use, please see Appendix for details:

### *Parameters*

### Sampling rate, sound speed in air, number of microphone and sources are specified. In addition, number of samples is calculated based on sampling rate and frame numbers.

### *Audio and time indexes*

### Waves are read by audio related function. In addition, time indexes are calculated by sampling rate.

fs = 16000;

c = 343;

nsrc = 3;

nmic = 3;

nsample = fs \* 12;

space = 0.25;

t = 0:1/fs:(12-1/fs);

waves = [audioread('mixture1.wav'), audioread('mixture2.wav'), audioread('mixture3.wav')];

Note that waves should contain samples.

## Calculate direction of 3 audio sources and generalize parameters

Repeat to specify spacing to apply different scene. Since only two pieces of audio are used, input of *BSS Locate* takes the mixture signals. Then, apply formula (4) to calculate angles, which is named by *theta*.

As a result, *tdoa* vector is round up by (meters)

In addition, *theta vector* is round up by (degrees)

waves\_mixture23 **=** **[**audioread**(**'mixture2.wav'**),**audioread**(**'mixture3.wav'**)];**

tdoa **=** bss\_locate\_spec**(**waves\_mixture23**,** fs**,** d**,** nsrc**,**'GCC-PHAT'**);**

theta **=** **[**asind**(**tdoa**(**1**)\***c**/**d**),** asind**(**tdoa**(**2**)\***c**/**d**),** asind**(**tdoa**(**3**)\***c**/**d**)];**

Note: unit of *tdoa* vector is meter and *theta* vector is degrees. Baseline of 0 degree is the vertical line in central.

## Beamforming Filtering

## Parametric Spectral Subtraction

# Conclusion and Future Work

It is challenge to get out of one channel signal from 3 sources while using microphone array system. Based on GCC-PHAT, it is useful to calculate TDOA for following procedures. In second stage, while it’s a trade off between performance of noise deduction and directional interference elimination. In previous attempt, several different ways have been involved to try the best performance, while we do not subjectively compare these methods in terms of SNR and error variation. It’s possible to test results and comparison. In future, we would like to have a deep look and research on TDOA of multiple microphones detection instead of only two.

Above all, the result is satisfied and English female audio is extracted effectively. Furthermore, it is audible to listen to the speech of English female while others do not have much impact on it. Furthermore, different methods of noise deduction are involved to pick the best one, while others are also remained.

References

1. Acoustic source localization. (2014, October 24). *Wikipedia*. Retrieved November 25, 2014, from <http://en.wikipedia.org/wiki/Acoustic_source_localization>
2. Phased Array System Toolbox. (n.d.). *Mathworks*. Retrieved November 25, 2014, from [http://www.mathworks.com/products/phased-array/](http://www.mathworks.com/products/phased-array/%20)
3. Documentation. (n.d.). *Mathworks*. Retrieved November 25, 2014, from <http://www.mathworks.com/help/phased/examples/acoustic-beamforming-using-a-microphone-array.html>
4. TDOA. (n.d.). *Wikipedia*. Retrieved November 29, 2014, from <http://zh.wikipedia.org/wiki/TDOA>
5. Multilateration (n.d.). *Wikipedia*. Retrieved November 29, 2014, from [http://en.wikipedia.org/wiki/Multilateration#Measuring\_the\_time\_difference\_in\_a\_TDOA\_system](http://en.wikipedia.org/wiki/Multilateration%23Measuring_the_time_difference_in_a_TDOA_system)
6. Blandin C, Ozerov A, Vincent E. Multi-source TDOA estimation in reverberant audio using angular spectra and clustering[J]. 2011.

Appendix

## Manual:

Following code is made in MATLAB and I keep the format of highlight which is easy for readers to read. There are several pieces of scripts; main entrance has been declared as main function which use other function with proper parameters. If you intend to run it, please place them separately with function name and keep them in common directory, then run the main script. Note that the provided audio files should be also placed with same name in main function (as default). The phased array toolbox is based on new version of MATLAB, please install at least 2014a version to access new features. (keep all the dependent function scripts in case to work properly)

## Code:

main Function

%% Extraction a Target Speech Signal from 3-channel Recordings

% Copyright (c) 2014-2015, Ottawa-Carleton Institute for Electrical

% and Computer Engineering of University of Ottawa

% Author: Liu Cheng & Hongyu Zou

% Student Number: 7486632 & 7493642

% Contact Email: lchen156@uottawa.ca & hzou047@uottawa.ca

% This script is based on bss\_locate\_spec.m written by Charles Blandin

% and Emmanuel Vincent in 2011, public example provided by MATLAB which

% is used as Phased Array Toolbox

% All work was modified by Liu Cheng & Hongyu Zou in Nov 2014 to

% apply related algorithms for actual application

%% General parameters setting

clc**;**

clear all**;**

close all**;**

fs **=** 16000**;** % sample frequency is set to 16kHz

c **=** 343**;** % propagation rate of sound in air (m/s)

nsrc **=** 3**;** % indicate number of source of three

nmic **=** 3**;** % similar, indicate number of microphones

nsample **=** fs **\*** 12**;** % total samples in all three mixture audio

space **=** 0.25**;** % space for each micphones is 25cm

t **=** 0**:**1**/**fs**:(**12**-**1**/**fs**);** % total time length in terms of samples

% read three mixture audio into waves

waves **=** **[**audioread**(**'mixture1.wav'**),** audioread**(**'mixture2.wav'**),**...

audioread**(**'mixture3.wav'**)];**

%% Calculate direction of 3 audio sources and generalize parameters

% Thanks for the work of Charles Blandin and Emmanuel Vincent, related

% MATLAB scripts are used and modified for calculation

% 'stft\_multi.m' is used as dependent function for bss\_locate\_spec function

% by work of Charles Blandin and Emmanuel Vincent

% distance between microphone 1 and microphone 3

% for applying calculation of TDOA

d **=** 0.25**;**

% read mixture audio

waves\_mixture23 **=** **[**audioread**(**'mixture2.wav'**),**audioread**(**'mixture3.wav'**)];**

% calculate tdoa using parameters above

tdoa **=** bss\_locate\_spec**(**waves\_mixture23**,** fs**,** d**,** nsrc**,**'GCC-PHAT'**);**

% store values of 3 angles into vector theta

theta **=** **[**asind**(**tdoa**(**1**)\***c**/**d**),** asind**(**tdoa**(**2**)\***c**/**d**),** asind**(**tdoa**(**3**)\***c**/**d**)];**

%% Beamforming Filtering

% Start by scratch

M**=**3**;** % Size of Rx Matrix

space**=**0.25**;** % distance between two sensors

t0**=**space**/**c**;**

a**=**0.985**;** % a parameter that controls the level of the point source noise relative to that of the spatially white noise

sita0**=**90**-**theta**(**3**);** % The direction of desired signal

sita1**=**90**-**theta**(**1**);** % The direction of interference1

sita2**=**90**-**theta**(**2**);** % The direction of interference2

y1**=**wavread**(**'mixture1.wav'**);**

y2**=**wavread**(**'mixture2.wav'**);**

y3**=**wavread**(**'mixture3.wav'**);**

Y1**=**fftshift**(**fft**(**y1**'));**

Y2**=**fftshift**(**fft**(**y2**'));**

Y3**=**fftshift**(**fft**(**y3**'));**

V**=[**Y1**;**Y2**;**Y3**];**

k**=**1**;**

I**=**eye**(**3**);** % white noise

signal**=**sita0**\***pi**/**180**;**

IR1**=**sita1**\***pi**/**180**;**

IR2**=**sita2**\***pi**/**180**;**

**for** f**=-**fs**/**2**:**fs**/(**192000**-**1**):**fs**/**2

v**=[**1**;**exp**(-**i**\***t0**\***2**\***pi**\***f**\***cos**(**signal**));**exp**(-**i**\***2**\***t0**\***2**\***pi**\***f**\***cos**(**signal**))];** % steer vector of desired signal

I1**=[**1**;**exp**(-**i**\***t0**\***2**\***pi**\***f**\***cos**(**IR1**));**exp**(-**i**\***2**\***t0**\***2**\***pi**\***f**\***cos**(**IR1**))];** % steer vector of interference1

I2**=[**1**;**exp**(-**i**\***t0**\***2**\***pi**\***f**\***cos**(**IR2**));**exp**(-**i**\***2**\***t0**\***2**\***pi**\***f**\***cos**(**IR2**))];** % steer vector of interference2

Sn**=(**1**-**a**)\***I**+**a**\*(**I1**\***I1**'+**I2**\***I2**');** % psn

WW**=**v**'\***inv**(**Sn**)\***v**;**

W**=**inv**(**Sn**)\***v**/**WW**;**

H**=**W**';**

B**(**k**)=**H**\***V**(:,**k**);**

k**=**k**+**1**;**

**end**

Z**=[**B**(**96001**:**192000**),**B**(**1**:**192000**)];**

X**=**ifft**(**Z**,**192000**);**

x**=**real**(**X**);**

wavwrite**(**x**,**16000**,**'Original\_0.985.wav'**);**

%% Parametric Spectral Subtraction

output**=**SSPARAB98**(**x**,**fs**,**2**);**

wavwrite**(**output**,**16000**,**'output.wav'**);**

% output1=SSBoll79(x,fs,2);

% wavwrite(output1,16000,'output1.wav');

%

% output2=MMSECohen2004(x,fs,2);

% wavwrite(output2,16000,'output2.wav');

%

% output3=MMSESTSA85(x,fs,2);

% wavwrite(output3,16000,'output3.wav');

%

% output4=MMSESTSA84(x,fs,2);

% wavwrite(output4,16000,'output4.wav');

%

% output5=WienerScalart96(x,fs,2);

% wavwrite(output5,16000,'output5.wav');

%

% output6=SSBerouti79(x,fs,2);

% wavwrite(output6,16000,'output6.wav');

%

% output7=SSMultibandKamath02(x,fs,2);

% wavwrite(output7,16000,'output7.wav');

%

% output8=SSScalart96(x,fs,2);

% wavwrite(output8,16000,'output8.wav');

%

% output9=SSPARAB98(x,fs,2);

% wavwrite(output9,16000,'output9.wav');

1. [↑](#footnote-ref-1)