Investigation of source localization and beamforming with a spherical microphone array

Steve Philbert
University of Rochester
sphilber@ur.rochester.edu

ABSTRACT

This project examines detection and localization of sound sources from a rigid spherical microphone array. The microphone array handles audio in two forms: raw, directly from individual elements; and encoded into ambisonics, a representation of audio as spherical harmonics. The detection and localization algorithms detect the number of active sources and determines their location. Raw audio utilizes time difference of arrival (TDOA), while ambisonic audio relies on a grid of ambisonic decoders. Source detection is based on an active source which is greater than a determined threshold, and sources should not occupy the same position around the array. The algorithms are run through simulations first, then optimized as plugins for use on a real-time digital audio workstation (DAW). A brain application runs parallel to the digital audio workstation which receives information on the number and location of active sources and translates this into information into beamforming plugins. The result is a series of audio tracks containing audio from spatially separated sources.

1. INTRODUCTION

The Eigenmike, shown in **Figure 1**, is a rigid spherical microphone with 32 elements. It has software to convert raw sound into an ambisonic field, then convert the ambisonic field into single channel beamformers. The beamformers require directional input (azimuth, and elevation), and a beam pattern (i.e. cardioid, hyper-cardioid, super-cardioid) [1]. The goal of the project is to implement an audio informed source localization algorithm that can detect audio sources, identify their location, and form a beam(s) in the direction of the audio. This operation should happen in real time.

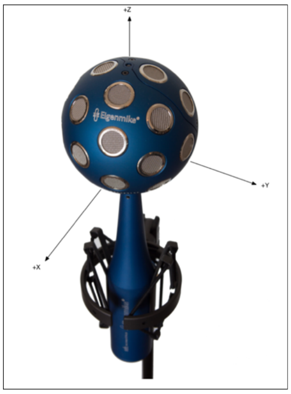


Figure 1. Eigenmike.

The spherical array is part of a microphone-centric scene where the microphone is the center of the scene, and sources are facing the microphone. The converse is a performance-centric with performers at the center and microphones pointing towards the performance [2]. Additionally, when a scene is performance-centric scene many microphones and microphone placement techniques are used to capture the scene; these include spot mics, microphone arrays, beamforming microphones both spherical and rectangular. Microphones pickup audio in with a line-of-sight method in their general direction; this contains both wanted sounds and unwanted sounds; this is a limiting factor in the ability to spatially separate sources.

Audio conferencing and concerts with small number of performers can utilize microphone-centric scenes. Audio conferencing where participants are sitting around a table with a microphone at the center, and concerts where musicians stand in a circle or semi-circle around a microphone. In these two scenes, the environments are relatively noise free and may contain some reverb. Also, the sound sources are distributed around the microphone with little to no overlap.

2. BACKGROUND

The spherical coordinate system is useful when working with spherical microphones and ambisonics. **Figure 2** shows the spherical coordinate system. Azimuth travels counterclockwise from 0° to 360°. Elevation is 0° degrees at the top, 90° on the horizontal, and 180° at the bottom. The eigenmike uses this coordinate system, however the speaker playback system used to generate datasets shifts the elevation by 90 degrees, resulting in +90° at the top, 0° on the horizontal and -90° on the bottom. Equations 1 and 2 show conversion between spherical and cartesian coordinate systems.

$$\begin{aligned} x &= r * \cos(elevation) * \cos(azimuth) \\ y &= r * \cos(elevation) * \sin\,(azimuth) \\ z &= r * \sin(elevation) \end{aligned} \qquad (1)$$

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ azimuth &= atan2(\frac{y}{x}) \\ elevation &= \operatorname{acos}\left(\frac{z}{r}\right) \end{aligned} \qquad (2)$$

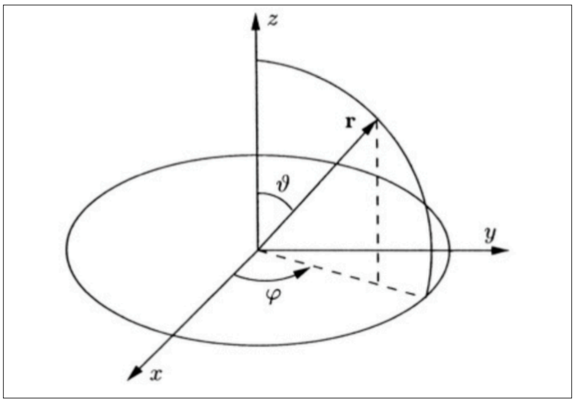


Figure 2. Spherical coordinate system.

2.1 Time difference of arrival (TDOA)

Time difference of arrival has a long history in communications systems especially radar. It extends now into wireless networks with respect to access points. For the scope of this it will deal with audio sources and microphones.

TDOA involves at least two detectors, and measuring the time difference between the signal arriving at each detector. This method works with two microphones, a rectangular array of microphones and spherical microphone arrays. [3, 4]

To determine the TDOA, first cross correlate the signals with each signal except its own signal. Next, find the index of the peak value of the cross-correlations. Use index to determine the lag value in samples from the cross correlation. It may be desirable to convert the lag value into seconds by dividing lag by the sample rate. It may be also desirable to find the lag in distance, divide lag by the sample rate, then multiply by the speed of sound.

2.2 Ambisonics

Spherical microphone arrays and spherical harmonics benefit from the same geometry, which makes ambisonics an easy method with spherical microphone arrays. Ambisonics uses the superposition of spherical harmonics to encode audio from the surface of a sphere into ambisonic channels, then decode ambisonic channels into audio on a surface of a sphere. [1, 5, 6, 7].

Meyer and Elko layout the formal derivation of ambisonics in their AES and ICASSP papers [5, 6]. **Figure 3** shows the spherical harmonics for zeroth, first, and second ambisonic orders.

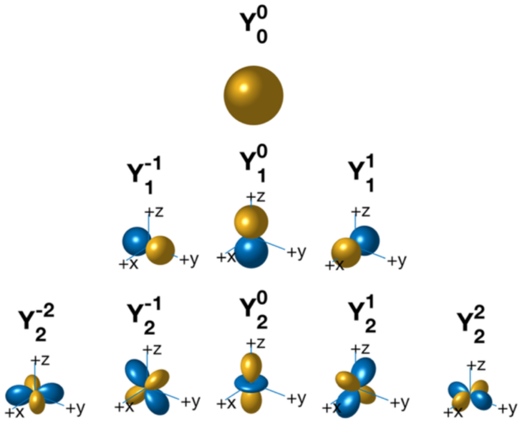


Figure 3. Spherical harmonics, shown to second order.

Equation 3 shows the ambisonic equations for zeroth order (W) and first order (Y, Z, X). These are provided in ambix format with ACN format and SN3D normalization [1, 7]. These equations translate the position of a source into ambisonic channels with spherical harmonics.

$$\begin{gathered} w = 1 \\ y = \sin(azimuth) * \cos(elevation) \\ z = \sin(elevation) \\ x = \cos(azimuth) * \sin(elevation) \end{gathered} \quad (3)$$

The eigenmike uses up to fourth order ambisonics, which requires 25 channels [1]. Each of the 32 eigenmike elements has a defined position on the surface of the sphere. The position data along with spherical harmonic equations are used to create an encoder matrix of size [32x25].

Audio can be decoded from ambisonic channels into positions using a similar method, the decoders could represent positions on the surface of the sphere. If these decoders were defined in a dense, evenly distributed grid, then it would be easy to determine the location when audio is present by examining the intensity at the decoders [8]. If 100 decoders with specified positions are used, then the decoder matrix would be size [25x100].

Ambisonics is good at estimating a bearing (azimuth, elevation); however, it is harder to estimate distance from a source [9, 10, 11]. One reason is the surface area of a sphere increases with by the square of the radius. A second reason is also based on geometry; localization is effective with triangulation but this requires detectors to be separated. On the spherical array, all the detectors are contained at the sphere. There's been work using reverberation to increase distance estimation with spherical microphone arrays; however, this is restricted to single source detection [4, 12, 14].

2.3 Sources

It is assumed there will be multiple sources of active and inactive audio in the scene. Most of the sources will stay in the same place, if there is movement, it shouldn't be very fast. A naive solution is to setup predefined fixed microphones; this would confine audio sources to a specific area; this method falls apart when sources move, or when sources become too close in proximity.

A better method is localization by the presence of audio which allows for movement of sources. However, this method falls apart when sources turn away from the microphone array, it's also difficult to distinguish between distance and loudness. Is the source louder because it increased in volume or louder because it is closer to the microphone? There's a question of active and non-active sources and movement. If a source becomes inactive and moves, should the system track this?

A more robust method is video informed source separation; this method would allow for tracking movement of sources, address the distance vs. loudness issue, and add address the active vs. non-active source.

In this project, audio informed source separation will be implemented. All the methods require spatial separation of sources with respect to the microphone array. As mentioned, with the geometry and line-of-sight source separation, all sources need to be spatially separated from each and orientated towards the microphone array.

3. DATASET

The dataset for testing and training was created in CSB505, spatial audio lab with the 27.2 speaker array. The eigenmike was placed at the center of the array. The speakers are installed in fixed locations. 25 speakers were used; three rings of eight speakers and one speaker directly over-head. Speakers' azimuth are 45 degrees apart, the elevation of upper and lower rings are +30 degrees, and -30 degrees while the center ring is at 0 degrees. The speaker array is capable of fourth order ambisonic and vector based amplitude panning (VBAP) playback; however, audio was played through the individual speakers without ambisonic or VBAP encoding. This was done to force a single source to a single speaker; whereas ambisonics produces sound from all 25 speakers and VBAP produces audio from three speakers. [13]

For convenience, the orientation of the speaker array (x-, y-, z-axis) matches the orientation of the microphone array (x-, y-, z-axis). The elevation of the eigenmike (0° to 180°) was offset to match the elevation for the speakers (90° to -90°). **Figure 4** shows a diagram of the relationship between microphone elements and speakers.

The audio in the data set was recorded using white noise, sine sweeps, single talkers, single instruments and purse sine tones. Each sound was individually recorded at each of the 25 positions for a duration of four seconds. Then multiple sounds were played at the same time through the speakers but not through the same speaker at the same time. Recordings were made with different levels of active sources two through eight. There are 40 recordings for each active source.

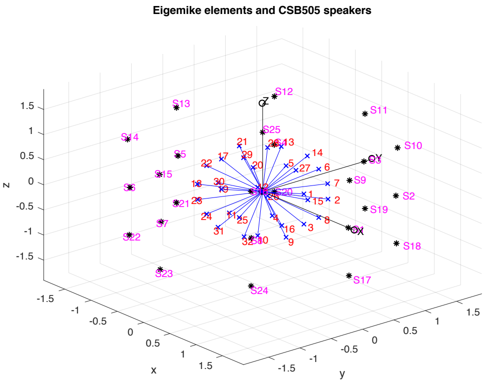


Figure 4. 3D representation of the environment; microphone elements are in red and blue. Speakers are in black and magenta.

4. SIMULATION

The simulation consists of analyzing two detectors, one a TDOA localization and the second is a ambisonic decoder grid.

4.1 Time Difference of Arrival (TDOA)

The first step is to collect audio at the elements of the microphone array. Then cross correlate the audio contained on the 32 channels. Find index of the largest peaks, then find lag of this index; this lag value is the difference in samples.

The next step is to determine the number of sources; however, running the simulation with multiple sources proved to be problematic at identifying the locations of multiple sources. If multiple sources were present, only one would be reported. This is due to the correlation between source and position. The number of sources is unknown and the positions are unknown making this a difficult task.

Once the number of sources is determined, for the TDOA is assumed one, then the localization process begins. In the localization, the elements with the maximum lag are identified, as the direction of the source. If one maximum exists, assume source is in the direction of that element. If two maximums exist, assume the source is in the direction of the vector normal to the bisection of those two elements. If three or more maximums, assume source is in the direction of the vector normal to a plane containing the three of these elements.

Figure 5 shows the cross correlation and the resulting lag. From the lag data, our position hits elements 1, 2, and 3, and approximately the same time. Table 1, shows the locations of microphone elements 1, 2, 3, the estimated direction vector from the plan formed by the three elements, and the ground truth location of position 1.

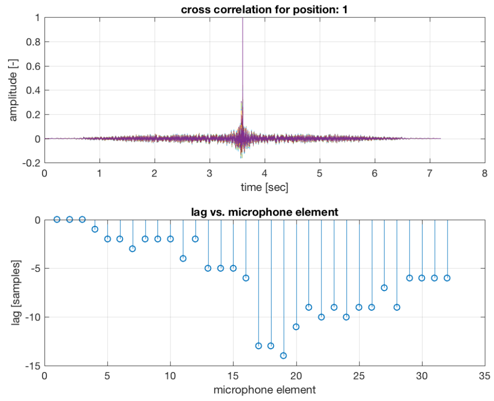


Figure 5. Cross-correlation and difference in lag for one position.

The localization method is a rough estimate with some error. **Figure 6** shows the accuracy of this method for each positon, the error is shown in degrees. The resolution of the lag is in samples, knowing the sample rate and speed of sound, this resolution may be converted into a quantized distance. One sample is 7mm and the radius of the eigenmike is 42mm; The distance of arrival from elements 1, 2, and 3 could be off by 14mm, when compared to the radius. With such a small radius, this quantization level leads to a high error. Two ideas to resolve this: increase the distance between elements for better triangulation or increase the sample rate to improve accuracy.

Location	azimuth	elevation
mic 1	0°	21°
mic 2	32°	0°
mic 3	0°	-21°
estimate	9.2°	0°
GT	0°	0°

Table 1. Table captions should be placed below the table.

Position 25 has a high error, it's located at the top of the sphere with elevation of 90°, the azimuth is defined as 0°; however at this point, the azimuth is not required. The estimated position for position 25 is azimuth of 180° and elevation of 89.6°. The elevation is close, but the estimate put the azimuth in the opposite direction, causing a high error.

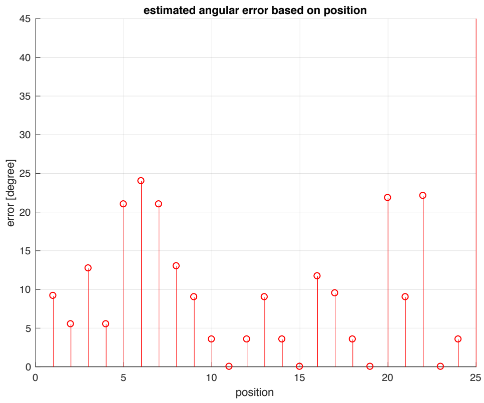


Figure 6. Accuracy of TDOA detectors.

4.2 Ambisonic decoder grid

The first step is to create a grid of ambisonic decoders, which are points distributed along the surface of a sphere, which contains 100 points. The next step is to construct an ambisonic decoder matrix, the size of this matrix is [25 x 100]. Audio is collected from the 32 elements, encoded into ambisonics with the supplied encoder matrix of size [32x25], this can be multiplied with the newly created ambisonic decoder, resulting in a 100-point representation of the surface of a sphere; a finer resolution than the 32 elements on the surface of the sphere

For each of the 100 points, the RMS value is recorded. A threshold can be set to pick peaks over a certain level. The peaks will be clustered together, each cluster represents an active sound source. If the sources are too close together, this is harder to differentiate between sources.

One the number of active sources is discovered, for each cluster, interpolate to fin the maximum rms value; this position corresponding to the value is where the source is estimated. **Figure 7** shows the ambisonic decoder grid and the rms level of each decoder with a threshold applied. The points above the threshold can be mapped into clusters on the surface of the sphere to determine the number of active sources and their positions.

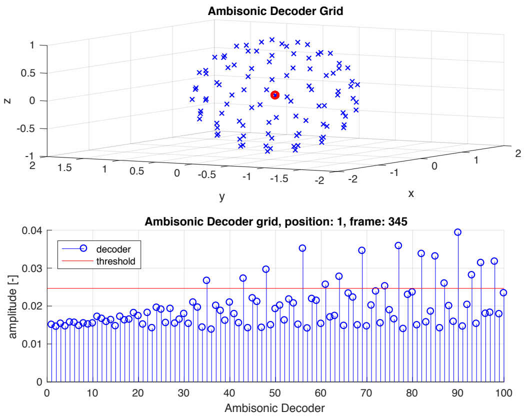


Figure 7. Ambisonic Decoder grid and detector thresholding.

5. REAL-TIME IMPLEMENTATION

This section describes the hardware implementation, DAW, Communication, and plugin configuration used to implement this system. **Figure 8** shows the overview and connections explained in this section.

5.1 Hardware

The Eigenmike connects to the Eigenmike interface box (EMIB) via a cat5 cable. The EMIB send audio on FireWire. A FireWire to thunderbolt adapter was used to bring audio into the mac.

5.2 DAW – Reaper Engine

Reaper was chosen as the real-time processing engine because it supports higher channel counts, integrates with Open Sound Control (OSC), has access to midi, and hosts the Eigenmike supplied AU plugins. Reaper runs in real-time by record enabling the tracks. Raw audio from the Eigenmike and the beamformers can be recorded. Additionally, a raw audio file can be loaded into Reaper and processed after the raw audio was recorded.

5.3 DAW – Reaper Engine

Max/MSP was chosen for its support of OSC and midi, and for its ability to process data. The detector plugins transit the number of active sources and their locations through midi to brain. The brain analyzes the information and determines how many beams and locations are required, then transmits this information to beamforming plugins in Reaper through OSC. This controls plugin parameters and polls the plugins for updated data.

5.4 Open Sound Control (OSC) and Midi Communication

There's a complex network of OSC and Midi messages communicating between Reaper and MAX/MSP to receiver data from plugins and send data to automate their controls. Figure XX shows the signal flow diagram. OSC and midi ports are configured in MAX/MSP and Reaper. Reaper requires an OSC template to determine the mapping of sources.

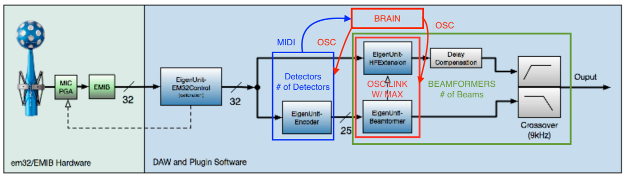


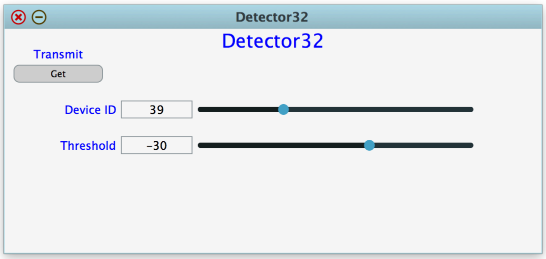
Figure 8. Big picture overview of signal flow from the microphone to the beamformers with plugins, OSC and midi information.

5.5 EigenUnit Plugins

EigenUnit plugins are provided by mh acoustics for use with the eigenmike. EigenUnit EM32Control: connects to eigenmike, set gain and calibration. EigenUnit Encoder: converts raw audio into ambisonic audio. EigenUnit Beamformer: creates a beam-former from ambisonic field. EigenUnit HF Ex-tension: Compensation for spatial aliasing in the geometry of the eigenmike which is used in conjunction with the beamformer. The Beamformer and HFExtension need to have the same positon; mh acoustics provides a JS plugin to synchronize the position data; however, this method only works for one beamformer per reaper project without modifying the JS script. The JS script was not used in this application, MAX/MSP can handle sending the position data to both the beamformer and the HFExtension. [1, 5, 6]

5.6 Detector32 plugin

The Detector32 VST3 plugin, shown in **Figure 9** has 32 input channels, to be fed from the eigenmike. It reports data via midi System Exclusive (SysEx) data to MAX/MSP. The transmit button when pressed will send device id, number of sources and their location through midi. The device id informs the brain what detector is used. The threshold informs the plugin to select audio detection which is larger than the threshold. All of these parameters are controlled by the brain through OSC. This plugin implements the time difference of arrival method with cross-correlation function in real time.



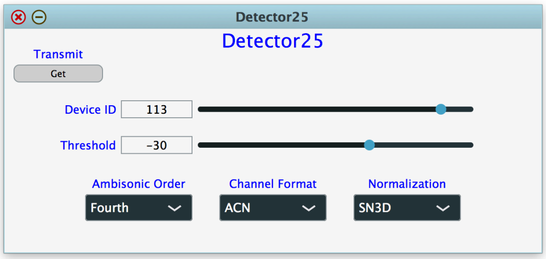
Device ID 39

Threshold -30

Figure 9. Detector32 plugin, implementation of time difference of arrival detector.

5.7 Detector25 plugin

The Detector25 VST3 plugin, shown in **Figure 10** has 25 input channels, to be fed from the ambisonic EigenUnits Encoder. It reports data via midi System Exclusive (SysEx) data to MAX/MSP. The transmit button when pressed will send device id, number of sources and their location through midi. The device id informs the brain what detector is used. The threshold informs the plugin to select audio detection which is larger than the threshold. Ambisonic parameters for order, normalization and format are controlled here too. All of these parameters are controlled by the brain through OSC. This plugin implements the 100-point ambisonic decoder grid in real-time.



Device ID 113

Threshold -30

Ambisonic Order: Fourth

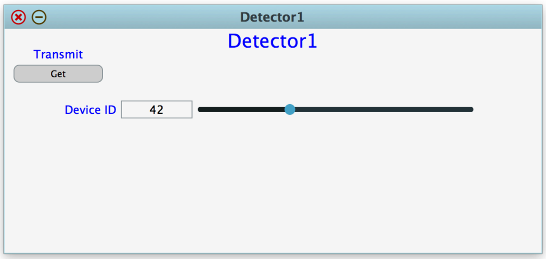
Channel Format: ACN

Normalization: SN3D

Figure 10. Detector25 plugin, implementation of ambisonic decoder grid detector.

5.8 Detector1 plugin

The Detector1 VST3 plugin, shown in **Figure 11** has one input channel. It reports data via midi System Exclusive (SysEx) data to MAX/MSP. The transmit button when pressed will send device id, and the RMS value of the channel to the brain. The device id informs the brain what detector is used. This is intended to measure the RMS value at each of the beams to ensure proper operation. It is also intended to measure the ambisonic Intensity or W-channel; this is an omnidirectional mono signal representing the intensity of the field, this can be used to determine the thresholds sent to the Detector32 and Detector25 plugins.



Device ID 42

Figure 11. Detector1 plugin, implementation of simple rms algorithm.

5.9 Actual Implementation

Unfortunately the full implementation was not able to be implemented in time for submission. The issue was completing and testing the implementation of the algorithms in the Detector32 and Detector25 plugins. The framework for the eigenmike and 8 beamformers, was implemented in Reaper and MAX/MSP with verification of Midi and OSC communication.

6. FUTURE WORK

Thresholding uses rms detection over the entire frequency range, band limiting these detectors may help improve results. Optimization of the ambisonic decoder grid to identify the best number of points and the best distribution of points.

This process did not utilize a trainable network for training on known sources or known positions; training may improve results. Additionally, video informed source localization would be a better method to track the locations of talkers and musicians; this works well in a spherical layout, as long as line-of-sight to the microphone is observed.

7. CONCLUSION

This work uses the microphone-centric spherical array which is very limiting in applications. Other array geometries may provide better results for certain applications, as well as combining microphone arrays to obtain unique positions and cover different angles.

The time difference of arrival detector works well for single sources. It is difficult to estimate multiple active sources with spherical array, because sound sources and element positions are correlated. Cross correlation gives the time of arrival, not the distance of arrival or source from which it arrived.

The Ambisonics detector separates the source and distance correlation by encoding location into spherical harmonics. The spherical geometry of the array makes this a good option. However, a large number of decoders is needed to accurately locate a source.

The implementation uses plugin provided with the Eigenmike; there are simpler methods to implement detector and beam forming circuits and projects. This was an overview of the possibilities of localization and beam forming.

REFERENCES

[1] mh Acoustics. Eigenmike Documentation. [Online]. Available: https://mhacoustics.com/download

[2] F. Wang, X Pan: "Acoustic Sources Localization in 3D Using Multiple Spherical Arrays," *Journal of Electrical Engineering & Technology*, Vol. 11, Number 3, May 2016.

[3] A. K. Tellakula, "Acoustic Source Localization using time delay estimation." MS. Dissertation. Dept. Eng. Indian Institute of Science Bangalore, India, August 2007.

[4] J. Scheuing, B Yang: "Disambiguation of TDOA Estimation for Multiple Sources in Reverberant Environments," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 8, November 2008.

[5] J. Meyer, G. Elko: "A highly scalable spherical Microphone array based on an orthonormal decomposition of the Soundfield," in *IEEE International Conference on Acoustics, Speech and Signal Processing.* 2002, Volume 2, pp. 1781-1784.

[6] J. Meyer, G. Elko: "Spherical microphone array for spatial sound recording," *presented at the 115th Audio Engineering Society Convention*, New York, NY, USA, 2013.

[7] C. Nachbar, et. al. "Ambix – A suggested Ambisonic Format" presented at *Ambisonics symposium,* Lexington, KY, USA. June 2011 [online] available http://iem.kug.ac.at/fileadmin/media/iem/projects/2011/ambisonics11_nachbar_zotter_sontacchi_deleflie.pdf

[8] P. Hack, "Multiple Source Localization with Distributed Tetrahedral Microphone Arrays." MS Thesis. Institute of Electronic Music and Acoustics at University of Music and Performing Arts at Graz University of Technology, Graz, Austria, 2015 Available: https://iem.kug.ac.at/fileadmin/media/iem/projects/2013/hack.pdf

[9] D. Charalampos, et. al., "Improved localization of sound sources using multi-band processing of ambisonic components," *presented at the 126th Convention,* Munich, Germany, May 2009

[10]A. Koutnya. et. al., "Source distance determination based on the spherical harmonics" *Mechanical Systems and Signal Processing,* Vol. 85, pp. 993–1004, 2017.

[11]Q. Haung, G. Zhang, K. Liu: "Near-Field Source Localization Using Spherical Microphone Arrays," *Chinese Journal of Electronics* Vol.25, No.1, Jan. 2016

[12]H. Liu, B. Yang, C. Pang: "Multiple sound source localization based on TDOA clustering and multi-path matching pursuit", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* presented New Orleans, LA, USA, March 2017

[13]P. Ville, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning," *Journal Audio Engineering Society*, Vol. 45, Issue 6, pp. 456-466, June 1997.

[14]H. Sun, et. al: "Joint DOA and TDOA estimation for 3D localization of reflective surfaces using Eigenbeam MVDR and spherical microphone arrays," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* March 2011