Audio Engineering Society

Conference Paper

Presented at the Conference on
Audio for Virtual and Augmented Reality
2018 August 20 – 22, Redmond, WA, USA

Real-Time Source-Tracking Spherical Microphone Arrays for Immersive Environments

Jonathan Mathews[1] and Jonas Braasch[1]

[1]*Rensselaer Polytechnic Institute*

Correspondence should be addressed to Jonathan Mathews (mathej4@rpi.edu)

ABSTRACT

Spherical microphone arrays have attained considerable interest in recent years for their ability to decompose three-dimensional soundfields. This paper details real-time capabilities of a source-tracking system composed of a beamforming array and multiple lavalier microphones. Using the lavalier microphones for source identification, a particle filter can be implemented to allow independent tracking of the orientation of multiple sources simultaneously. This source identification and tracking mechanism is utilized in an immersive lab space. In conjunction with networked audiovisual equipment, the system can generate a real-time virtual representation of sound sources for a more dynamic telematic experience.

1 Introduction

A diverse selection of literature exists on the topic of directionally filtering signals from higher order spherical microphone arrays. These deliver a variety of methods to perform soundfield analysis or source tracking with increasing precision and flexibility. The research effort outlined in this paper aims to develop a multiple-source-tracking system using modern techniques capable of providing relatively low-latency tracking data to a full telematic environment. This paper also demonstrates that such a system is fit to run on modern, consumer-grade equipment.

2 Localization

There are three major aspects to the method of localization and source identification in this application. The first is the beamforming algorithm, a delay-and-sum method which improves robustness to sensor noise while remaining both mathematically and computationally simple to execute, especially in the Spherical Harmonics Domain (SHD). The second aspect is the filtering mechanism, which treats a collection of generated beams as a set of particles, and uses an uncertainty model to iteratively predict and update a source's orientation over successive frames. Finally, the filtering mechanism is wrapped into a detection framework, which monitors source activity and allows for discrimination between multiple sources, while reducing com-

putational load.

2.1 Beamforming

Before beamforming, the signals from the microphone array are encoded in to the SHD using the Spherical Harmonics Transform (SHT). This improves the computational efficiency of beamforming down the line by taking advantage of the orthonormal spherical basis functions to describe the spatial behavior of the soundfield [1]. For a spherical array with Q sensors, with their orientation given by $\Omega_q = (\phi_q, \theta_q)$, a short-time Fourier transform of the input signals is computed first, then the SHT is performed by:

$$p_{nm}(k,a) = \frac{4\pi}{Q} \sum_{q=1}^{Q} p(k,a,\Omega_q) Y_n^m(\Omega_q)^*, \quad (1)$$

where $p(k,a,\Omega_q)$ is the frequency-domain representation of the audio signals, a is the radius of the sphere, and $Y_n^m(\Omega_q)$:

$$Y_n^m(\Omega) = \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}} P_{nm} cos(\theta) e^{jm\phi} \quad (2)$$

are the Spherical Harmonics coefficients, oriented for each capsule position, governed by degree n, order m, angular orientation Ω and the Legendre function P_{nm}. This operation can be represented in matrix form as:

$$\mathbf{p_{nm}} = \frac{4\pi}{Q} \mathbf{Y_q^H p}, \quad (3)$$

where $\mathbf{p} \in \mathbb{C}^{(Q \times S)}$ represents the audio signals, and $\mathbf{Y_q} \in \mathbb{R}^{(N+1)^2 \times Q}$ is the matrix of harmonic components. The SHD representation of the array is order-limited based on $Q = (N+1)^2$, where Q is the number of microphones, and N is the maximum ideally attainable order before spatial aliasing occurs. For further details of the SHT for acoustic arrays and its parameters, refer to [1, 2, 3, 4].

The beamformer, or directional filter, is a simple delay-and-sum model, designed to align phase of the input signals in the direction of interest. In the SHD, the weights are described by

$$w_{nm} = d_n(kr) \frac{Y_n^m(\Omega_l)}{b_n(kr)}, \quad (4)$$

where d_n is the set of axis-symmetric delay-and-sum beamforming weights, as derived in [2]. $b_n(k,r)$ represents the modal behavior of the spherical surface at the orientation described by $Y_n^m(\Omega_l)$ using spherical harmonics. These are applied to the transformed array signals to create the directionally-filtered array output:

$$\mathbf{y} = \mathbf{w} \circ \mathbf{p_{nm}}. \quad (5)$$

Multiple beams can be generated and applied to the audio data with this method. The output is then a matrix of spherical harmonic signals that have been directionally filtered according to multiple orientation angles.

White-Noise Gain is a metric of array robustness to sensor self-noise, and can be determined by taking the ratio of the array input to the array output. By maximizing the ratio, the optimal weights for maximum WNG for a SHD beamformer can be derived,

$$d_n = \frac{|b_n(kr)|^2}{\sum_{n=0}^{N} \frac{2n+1}{4\pi} |b_n(kr)|^2}, \quad (6)$$

which is analogous to the delay-and-sum beamformer, as shown in [5]. Although other algorithms for directional filtering exist in the literature, this method is preferable for its directional and computational performance, as well as its simplicity [6, 7].

2.2 Filtering Algorithm

The generated beams are incorporated into a filtering algorithm to estimate source position over time. The energy sum of the array output is taken, then normalized across orientation angles to generate a pseudolikelihood distribution. A recursive Bayesian estimator is used to estimate source position based on this likelihood function as well as a transition model which estimates position change over time, and prior output.

The conditional density estimating source position given array measurements is described by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \quad (7)$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}), \quad (8)$$

where $\mathbf{x}$ is the node corresponding to source position, and $\mathbf{y}$ is the data obtained from measurements. $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is the posterior density which describes the probability of a source position given a collection of measurements to step t. $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition density which governs movement of the source. Finally, $p(\mathbf{y}_t|\mathbf{x}_t)$ is the likelihood. Recursive approximations of the posterior density are performed using a sequential Monte Carlo process on a set of particles which are generated from the array output as described above [8, 9]. This process of recursive estimation is outlined as follows:

Given an initial uniformly-weighted particle distribution, the source state represented by $\mathbf{x}^{(i)}$, with $i \in 1 \ldots N$ being the set of particles,

1. Generate a likelihood function based on measurements obtained.

   In lieu of a true likelihood function, a pseudolikelihood is generated via

   $$f(\mathbf{x}_t, \mathbf{y}_t) = \max \mathbf{y}_t(\Omega), \quad (9)$$

   where Ω is the set of vectors governing state orientation ($\Omega^{(i)} = (\theta^{(i)}, \phi^{(i)})$. This takes advantage of the fact that the measurements from generated beams imply their own distribution of the source location estimate [10]. And therefore, the particle orientation is observed where the measurement values are at maximum.

2. Apply normalized values from the likelihood function to the particle weights.

   The likelihood values are converted to weights by normalizing their values to sum to unity.

   $$w_t = \frac{f(\mathbf{x}, \mathbf{y}_t)}{\sum f(\mathbf{x}, \mathbf{y}_t)} \quad (10)$$

3. Estimate source position

   The weighted particles are applied to the orientation set to produce the source location estimate.

   $$L_t = \sum_{i=1}^{N} w_t^{(i)} \Omega^{(i)} \quad (11)$$

4. Predict a new set of particles using the estimated source location and a model for source movement The weighted particles are resampled according to a selection-with-replacement scheme that prioritizes particles with large weighting values. There are a variety of dynamic models that are suitable for this work [8, 11, 12]. In this case, a simple Gaussian distribution was used, with the parameters determined heuristically.

5. Save the new particle orientation set for processing the next frame of measurement data

The process is then repeated for the next frame of data using the resampled particles.

2.3 Detection Framework

The framework governing this process allows discrimination of multiple sources. Lavalier microphones are used to identify active sources on a per-frame basis. Based on the active source, a corresponding set of particles are updated to reflect its estimated orientation. Finally, a time-based limit is imposed on the orientation of the particle sets to account for source motion during periods of inactivity.

The energy-sum of each lavalier microphone within a given time frame is determined by

$$E_{\text{lav}} = \sum_{n=1}^{N} s(n)^2. \quad (12)$$

Where $n \in 1 \ldots N$ are samples, and $s(n)$ is the signal from the lavalier. The log difference of the lavalier signals is compared, and identifies whether a single source is active, multiple sources are active, or no sources are active, based on whether the difference result exceeds an experimentally determined threshold.

From this identification, the system will activate a set of particles for prediction and update only if its corresponding source is solely active. If both sources are either active or inactive, the system defaults to a holding state, storing the last set of computed particles, and activating a timer which will reset the particles for all sources to initial weight and orientation distributions once a given time interval has elapsed. Similarly, while a single source is active, reset timers are engaged for all inactive sources. This reset process is included to

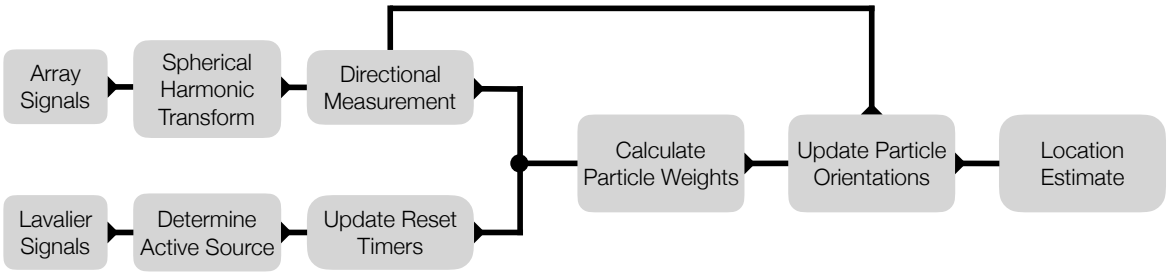

Fig. 1: Global flow diagram of the detection and tracking system for a single frame of data.

improve tracking accuracy over relatively long periods of inactivity as seen in typical conversational scenarios. A diagram of this detection and activity analysis framework is shown in Figure 1.

For analysis purposes, the absolute position of the sources are recorded and compared with the estimate using

$$\varepsilon_t = |L_t - \hat{L}_t| \tag{13}$$

Where ε_t is the angular deviation for time frame t. This is averaged over the runtime of each test to produce the mean error, quantifying the degree of deviation from the true source position.

3 Results

A test of this system was performed at the Rensselaer Polytechnic Institute CRAIVE-Lab (Collaborative Research-Augmented Immersive Virtual Environment) - a large, open room approximately 223m^2 and with an reverberation time of 0.4s. Two lavalier microphones were attached to two sound sources simulating a conversation between two male speakers, with speaking intervals varying between 2 and 4 seconds, and consistent intervals of either concurrent or sequential speech. The first source maintained a stationary position at 2.3m, while the second moved along an arc of approximately πrad around the array, also at a distance of 2.3m.

The system was compiled as an external object in Max 7 to take advantage of Max's real-time audio processing capabilities. Each audio frame consisted of 512 samples with a 50% overlap, and a Hamming window applied. Audio was obtained via three slaved Focusrite interfaces with a sample rate of 48ksps and 16-bit depth. Computation was performed on a consumer-grade laptop. Plots of the test were generated using a functionally identical system written in MATLAB.

The plot of position over time for the two sources and their estimates are displayed in Figure 2. Accuracy was calculated to be 0.07rad for concurrent speakers, and 0.1rad for sequential speakers via average angular deviation. Computational speed of the system was logged in both Max and MATLAB - average processing time per-frame was 1.1ms for Max, or about 890 audio frames per second.

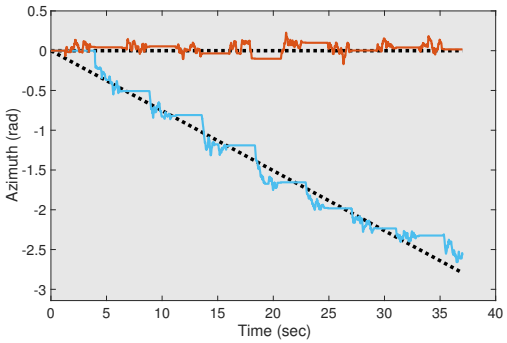

Fig. 2: Plot of the average estimated position over azimuth of two sequentially speaking male speakers, one stationary (red) and one moving (blue), with ground truth represented by the dashed black lines.

4 Discussion

The low latency of this system justifies a low threshold for dropped frames (i.e. frames which bypass analysis due to source confusion). Since typical speech sources

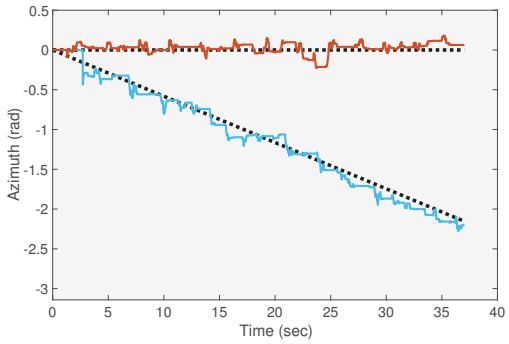

Fig. 3: Plot of the average estimated position over azimuth of two concurrently speaking male speakers, one stationary (red) and one moving (blue), with ground truth represented by the dashed black lines.

produce large dynamic changes in very short time intervals, tracking of even concurrent speakers is possible with fast enough frame-by-frame analysis.

It should also be noted that the system was limited to azimuth-only detection for ease of analysis and plot generation. However, extension to full spherical coverage is a relatively trivial exercise involving uniform spherical coverage instead of uniform circular for the initial particle distribution, and dynamic motion in both azimuth and elevation versus simply azimuth.

The higher error value seen in the sequential source case can mostly be attributed to the continued motion of the moving source. Since the default behavior of the system is to maintain the orientations of the last frame of particles verified by measurement, there is a period of tracking loss, followed by a brief period of reacquisition as the system registers new activity.

By compiling the system for use in Max, integration with other technical tools becomes a viable option. Currently, extensions of the system are capable of streaming audio and tracking data to other computers, web-based applications, and other devices in the environment, such as tracking cameras and networked lighting systems. This ease of integration aids research into multi-sensor tracking at the environment-level.

5 Summary

This research effort demonstrates a practical framework for multiple source discrimination with a beamforming spherical microphone array. The ability to generate accurate tracking with high frame rates on consumer-grade hardware is important for the proliferation of this type of acoustic analysis in the commercial sphere. Although many refinements can be made to improve the computational efficiency, tracking behavior, hardware robustness, etc., the simplicity and ease of implementation of this system indicates its usefulness in a variety of practical scenarios where real-time performance is important.

Fig. 4: Tracking and integration test. The location estimate of the author is broadcast to a lighting array and browser-based application, visually highlighting the author's position.

6 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant #1631674, the RPI Cognitive and Immersive Systems Laboratory, and the RPI Humanities, Arts, and Social Sciences Fellowship.

References

[1] Meyer, J. and Elko, G., "A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield," in *2002 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. II–1781–II–1784, 2002.

[2] Rafaely, B., *Fundamentals of Spherical Array Processing*, Springer, Berlin, Germany, 1 edition, 2015.

[3] Li, Z. and Duraiswami, R., "Flexible and Optimal Design of Spherical Microphone Arrays for Beamforming," *IEEE Trans. Audio, Speech, Language Process.*, 15(2), pp. 702–714, 2007.

[4] Abhayapala, T. D. and Ward, D. B., "Theory and design of high order sound field microphones using spherical microphone array," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. II–1949–II–1952, 2002.

[5] Rafaely, B., "Phase-mode versus delay-and-sum spherical microphone array processing," *IEEE Signal Processing Letters*, 12(10), pp. 713–716, 2005.

[6] Cigada, A., Ripamonti, F., and Vanali, M., "The delay & sum algorithm applied to microphone array measurements: Numerical analysis and experimental validation," *Mechanical Systems and Signal Processing*, 21(6), pp. 2645 – 2664, 2007.

[7] Mucci, R., "A comparison of efficient beamforming algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(3), pp. 548–558, 1984.

[8] Vermaak, J. and Blake, A., "Nonlinear filtering for speaker tracking in noisy and reverberant environments," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 5, pp. 3021–3024 vol.5, 2001.

[9] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, 50(2), pp. 174–188, 2002.

[10] Ward, D. B. and Williamson, R. C., "Particle filter beamforming for acoustic source localization in a reverberant environment," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pp. II–1777–II–1780, 2002.

[11] Lehmann, E. A. and Williamson, R. C., "Particle Filter Design Using Importance Sampling for Acoustic Source Localisation and Tracking in Reverberant Environments," *EURASIP J. Appl. Signal Process.*, 2006, pp. 168–168, 2006.

[12] Doucet, A. and Johansen, A. M., "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, 12(656-704), p. 3, 2009.