MUSIC-BASED SOUND SOURCE LOCALIZATION AND TRACKING FOR TASKS 1 AND 3

Kazuhiro Nakadai[1,2], *Katsutoshi Itoyama*[2], *Kotaro Hoshiba*[3], *and Hiroshi G. Okuno*[4]

1. Honda Research Institute Japan Co., Ltd.,
8-1 Honcho, Wako, Saitama 351-0188, Japan
2. School of Engineering, Tokyo Institute of Technology,
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
3. Faculty of Engineering, Kanagawa University,
3-27-1 Rokkakubashi, Kanagawa-ku, Yokohama, Kanagawa 221-8686, Japan
4. Graduate Program for Embodiment Informatics, Waseda University,
2-4-12 Lambdax Bldg. 301, O-kubo, Shinjuku-ku, Tokyo 169-0072, Japan

ABSTRACT

This paper presents our results for two microphone arrays such as "robot head" and "eigenmike" in tasks 1 and 3 of the LOCATA Challenge. We basically used multiple signal classification (MUSIC) based on generalized eigenvalue decomposition with geometrically-calculated steering vectors which is implemented in the open source robot audition software HARK (Honda Research Institute Japan Audition for Robots with Kyoto University). To deal with the tasks, we added two additional procedures; 1) We used voice activity detection (VAD) based on a zero cross rate and power thresholding instead of using HARK-based VAD by peak power thresholding on the MUSIC spectrum to solve a mismatch problem produced by the geometrically-calculated steering vectors. 2) We performed Kalman filter based tracking which takes dynamic changes of the number of sound sources into account. We constructed a sound source localization system by combining an online MUSIC module in HARK and other offline modules of VAD and tracking with MATLAB. We also proposed three evaluation metrics, and analyzed the localization results for the provided evaluation data set using the metrics to clarify the characteristics of the proposed system. We showed that the performance of sound source localization is maintained when MUSIC is performed only once in every 20 frames. This achieved real-time processing for the microphone array of "robot head."

Index Terms— Sound source localization, MUSIC, voice activity detection, sound source tracking

1. OVERVIEW OF THE PROPOSED METHOD

Fig. 1 illustrates the diagram of the proposed system, which consists of sound source localization, frame-based voice activity detection (VAD), and sound source tracking. The following sections describe the algorithms used in these components.

1.1. Sound Source Localization

Sound source localization is one of the most primary functions in the field of signal processing. We have been developing several algorithms based on multiple signal classification (MUSIC). The original MUSIC algorithm is based on standard eigenvalue decomposition (hereafter, SEVD-MUSIC) [1], and it can produce

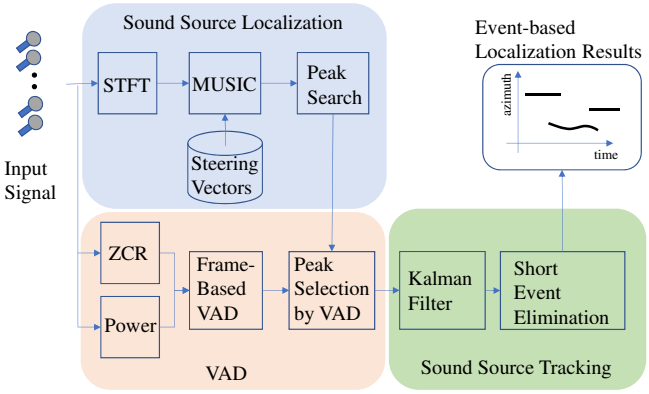

Figure 1: The diagram of the proposed system.

a high peak for every sound source direction. Since the original MUSIC deteriorates when the power of noise is higher than that of a target signal, it is extended to solve this problem by noise-whitening and generalized eigenvalue decomposition (hereafter, GEVD-MUSIC) [2].

Let $\boldsymbol{X}(\omega, f)$ be the observed signal vector at the f-th frame at the ω-th frequency bin generated from multi-channel input signals by applying short-time Fourier transform (STFT) with the frame window length W, the window shift length N, and the FFT(fast Fourier transform) length F. The correlation matrix $\boldsymbol{R}(\omega, f)$ is obtained from $\boldsymbol{X}(\omega, f)$ by,

$$\boldsymbol{R}(\omega, f) = \frac{1}{T_R} \sum_{\tau=f}^{f+T_R-1} \boldsymbol{X}(\omega, \tau)\boldsymbol{X}^*(\omega, \tau), \quad (1)$$

where T_R is the number of frames for temporal integration to calculate the correlation matrix.

SEVD-MUSIC simply performs standard eigenvalue decomposition for the correlation matrix, which is defined as,

$$\boldsymbol{R}(\omega, f) = \boldsymbol{E}(\omega, f)\boldsymbol{\Lambda}(\omega, f)\boldsymbol{E}^*(\omega, f), \quad (2)$$

where $\boldsymbol{\Lambda}(\omega, f)$ is a matrix whose diagonal elements are eigenvalues in descending order. $\boldsymbol{E}(\omega, f)$ is a matrix consisting of eigenvectors.

On the other hand, GEVD-MUSIC takes another input, that is, a noise correlation matrix $\boldsymbol{K}(\omega, f)$, and it is estimated using the first T_N frames of the input signal as,

$$\boldsymbol{K}(\omega) = \frac{1}{T_N} \sum_{\tau=1}^{T_N} \boldsymbol{X}(\omega, \tau) \boldsymbol{X}^*(\omega, \tau) \,, \tag{3}$$

After the whitening process is performed by calculating $\boldsymbol{K}^{-\frac{1}{2}}(\omega)\boldsymbol{R}(\omega, f)\boldsymbol{K}^{-\frac{1}{2}}(\omega)$, GEVD performs eigenvalue decomposition defined as,

$$\boldsymbol{K}^{-\frac{1}{2}}(\omega)\boldsymbol{R}(\omega, f)\boldsymbol{K}^{-\frac{1}{2}}(\omega) = \boldsymbol{E}(\omega, f)\boldsymbol{\Lambda}(\omega, f)\boldsymbol{E}^*(\omega, f), \tag{4}$$

The MUSIC spatial spectrum $P(\omega, \psi, f)$ is calculated using steering vector $\boldsymbol{G}(\omega, \psi)$ as,

$$P(\omega, \psi, f) = \frac{|\boldsymbol{G}^*(\omega, \psi)\boldsymbol{G}(\omega, \psi)|}{\sum_{m=L+1}^{M} |\boldsymbol{G}^*(\omega, \psi)\boldsymbol{e}_m(\omega, \psi)|}, \tag{5}$$

where ψ is a sound source direction, and L is the number of target sound sources. $\boldsymbol{e}_m$ shows the m-th eigenvector included in $\boldsymbol{E}$.

The first L eigenvectors in $\boldsymbol{E}$ correspond to target sound sources, and others are related to noise sources. When ψ shows a sound source direction, an inner product of $\boldsymbol{G}^*(\omega, \psi)$ and $\boldsymbol{e}_m$ goes to 0 for every $m \geq L + 1$ because signal and noise vectors are orthogonal to each other in the signal space spanned by $\boldsymbol{E}_l$. This means that the denominator of Eq. (5) becomes 0, and thus a sharp peak is formed for the sound source direction ψ in $P(\omega, \psi, f)$.

$\boldsymbol{G}$ can be obtained precisely from measurement-based impulse responses. However, such measurements are not available, and thus, we calculated $\boldsymbol{G}$ using the geometrical relationship between microphones and sound sources by assuming the free acoustic field.

After that, $P(\omega, \psi, f)$ is averaged on ω denoted as,

$$\bar{P}(\psi, f) = \frac{1}{\omega_H - \omega_L + 1} \sum_{\omega=\omega_L}^{\omega_H} P(\omega, \psi, f), \tag{6}$$

where ω_H, and ω_L are the highest and lowest boundary of the frequency bin, respectively.

Finally, $\bar{P}(\psi, f)$ is sent to frame-based VAD. Note that eigenvalue decomposition (EVD) in Eq. (2) or (4) is computationally expensive, and thus, in actual implementation, Eqs. (2), (4), (5), and (6) are performed once every T_E frames. Such implementation is avaliable in HARK (Honda Research Institute Japan Audition for Robots with Kyoto University)[1] [3]. We used these algorithms to deal with tasks for the LOCATA challenge.

1.2. Frame-based Voice Activity Detection

Sound source direction is usually estimated as ψ which has a peak exceeding a threshold P_{th} in $\bar{P}(\psi, f)$. Since this thresholding decides whether a sound exists or not at every frame, it performs frame-base VAD in addition to the estimation of sound source direction. However, this normal thresholding process causes a problem when $\boldsymbol{G}$ has a mismatch with the actual steering vectors. As mentioned above, it is inevitable that there is a mismatch between the calculated and the actual steering vectors. It produces errors in a MUSIC spectrum generated from the input acoustic signal. Because, in our preliminary experiments, the errors affected MUSIC spectrum generation in low power signal periods like silent parts compared to that in high power periods where a target signal exists, we did not use this thresholding on the MUSIC spectrum. Since evaluation data in tasks 1 and 3 do not include so much noise, we decided to perform a conventional thresholding algorithm based on a zero cross rate (ZCR) and frame power rate (FPR) for frame-based VAD. When $s_i(t)$ is an input signal of the i-th channel at time t, ZCR at the f-th frame, $Zr(f)$, is defined by,

$$Zr(f) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{t=f \cdot N}^{f \cdot N + W - 1} z(t, i) \tag{7}$$

$$z(t, i) = \begin{cases} 1 & s_i(t) \cdot s_i(t+1) < 0 \\ 0 & otherwise. \end{cases} \tag{8}$$

FPR at the f-th frame, $Pr(f)$, is defined by,

$$Pr(f) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{t=f \cdot N}^{f \cdot N + W - 1} |s_i(t)| \,. \tag{9}$$

Frame-based VAD is, then, defined by,

$$Vad(f) = \begin{cases} 1 & Zr(f) > \mathrm{Th}_{zr} \text{ and } Pr(f) > \mathrm{Th}_{pr} \\ 0 & otherwise. \end{cases} \tag{10}$$

Once frame-based VAD is performed, at the voice activity periods, a peak in $\bar{P}(\psi, f)$ is extracted as ψ. The peak value with ψ is sent to sound source tracking.

We implemented the VAD algorithm with MATLAB to be performed as an offline post-process of MUSIC-based localization, although the algorithm itself can be implemented as online processing because it includes calculation of the zero cross rate and power of the input signal.

Table 1: Parameters of the system with values for the LOCATA Challenge

General Parameters	
Frame window length (W)	480 points
Window shift length (N)	160 points
FFT length (F)	512 points
Parameters for Sound Source Localization	
Localization algorithms	*SEVD* (Eq. (2)), *GEVD* (Eq. (4))
EVD frequency (T_E)	every *1, 10, 20, 30, 40, 50, 60, 70* frames
Num of sources (L)	1
#frames for correlation matrix (T_R)	50 frames
#frames for noise correlation matrix (T_N)	50 frames
Frequency ranges (ω_L and ω_H)	*125–2,800 Hz, 125–4,000 Hz,125–7,500 Hz 500–2,800 Hz, 500–4,000 Hz*
Steering vectors ($\boldsymbol{G}$)	calculated at 5° intervals in azimuth without elevation
Parameters for Frame-based VAD	
Threshold for ZCR (Th_{zr})	1.3×10^{-2}
Threshold for FPR (Th_{pr})	3.2×10^{-4}
Parameters for Sound Source Tracking	
pause length (PL)	0.2 sec
minimum length (ML)	0.1 sec

[1] https://www.hark.jp/

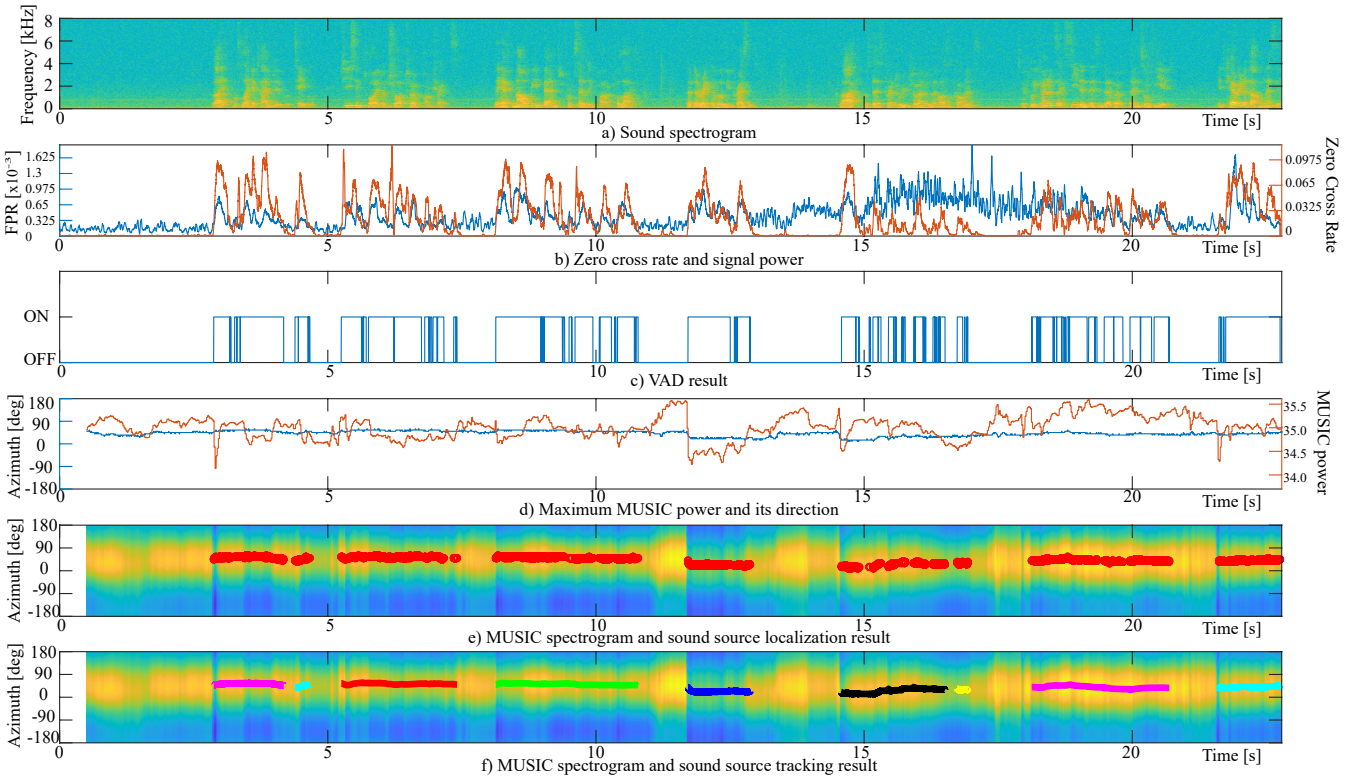

Figure 2: A result for "task3_recording1_eigenmike" recorded with the "eigenmike" microphone array in task 3: a) sound spectrogram obtained by STFT, b) frame power rate with a blue line using a Y-axis on the left side and zero cross rate with a red line using another Y-axis on the right side, which are calculated by Eqs. (7) and (9), respectively, c) frame-based VAD results estimated by Eq. (10), d) a sequence of peak values in MUSIC spectrum with a red line using a Y-axis on the right side and the corresponding azimuth values with a blue line using a Y-axis on the left side, e) MUSIC spectrogram and the selected peaks with red circles according to the VAD result, f) MUSIC spectrogram and sound source tracking results with colored lines by Kalman filtering.

1.3. Sound Source Tracking

The frame-based VAD has some parameters, which are basically manually tuned. However, it is difficult to have the optimal parameters for all evaluation data. This indicates that the VAD is not perfect, and thus we further performed Kalman filtering to form a sound event from the frame-based VAD results, and the Kalman filtering can also provide sound source tracking. The tracking is applied only for the azimuth direction using a simple Kalman filter based on linear motion of uniform acceleration in the time domain. Since the number of sound sources is dynamically changing, 0 or 1 in tasks 1 and 3, we introduced two procedures to deal with the dynamic changes to design the Kalman filtering. One is that we introduced a parameter called pause length (PL) to decide if a source is terminated, that is, when no observation is associated with a tracked source more than PL periods, the source is terminated. The other is that when no source is found to be associated with an observation, a new source is generated with the observation. After that, short sound events among the formed ones are eliminated as outliers, that is, when the length of the source is less than minimum length (ML), the source is eliminated. This tracking algorithm is implemented with MATLAB as another offline post-process of frame-based VAD. The Kalman filtering can be implemented as online processing, but a delay of the length of the minimum sound event is unavoidable in the elimination of short events. When we have an integrated system including all processes mentioned above, it can be implemented to work incrementally with the delay which is required in the elimination.

1.4. An example of sound source localization

Fig. 2 depicts an example of sound source localization results with the proposed system for a file named "task3_recording1_eigenmike" included in task 3 of the evaluation data set. Fig. 2a) illustrates sound spectrogram obtained after STFT in Sound Source Localization. After that, MUSIC is performed and MUSIC spectrogram is outputted as background color maps in Fig. 2e) and 2f). From the MUSIC spectrogram, the peak value is extracted with its direction at every frame shown as Fig. 2d). In VAD, ZCR and FPR are calculated, which is demonstrated in Fig. 2b). By integration of ZCR and FPR, VAD is extracted as Fig. 2c). According to the VAD result and the extracted peaks in Fig. 2d), peaks are selected as red circles in Fig. 2e). Note that these circles are not connected in a temporal direction. Finally, Kalman filtering performed for the red circles, and sound events are formed as lines in Fig. 2f).

The system has several parameters summarized in Table 1. The parameter tuning is an important factor to attain high performance and real-time processing. The cells with multiple values show the parameters we explored for the tasks in the LOCATA challenge,

which will be discussed in the next section.

2. EVALUATION

The tests were performed by changing values of the selected major parameters, and the obtained results were analyzed with discussions. By assuming that the results with a parameter set (GEVD, T_E = 1, ω_L = 125, ω_H = 4,000 in Tab. 1) are ground truths (hereafter, GTP set), three metrics such as localization accuracy (L_{ACC} [%]), coverage accuracy (T_{ACC} [%]), and angle difference (A_{DIF} [deg]) are defined by,

$$L_{ACC} = \frac{C_N - I_N}{N_N} \quad (11)$$

$$T_{ACC} = \frac{C_t - I_t}{N_t} \quad (12)$$

$$A_{DIF} = \frac{1}{|T_c|} \sum_{n \in T_c} |d(n) - d_G(n)| \quad (13)$$

where C_N and I_N the number of matched and extra detected sound events, and N_N shows the total number of sound events detected with the GTP set. C_t, I_t, and N_t represent the total durations of C_n, I_n, and N_n, respectively. T_c is a set of the overlapping time samples included in the matched sound events, of which duration corresponds to C_t. $d(n)$ and $d_G(n)$ represent azimuth angles of the estimated sound event and ground truth at the n-th time sample. Note that all data were processed on a laptop computer with an Intel Core i7-5700HQ 2.7 GHz CPU.

2.1. Comparison of localization methods

Fig. 3 shows the averaged scores of L_{ACC}, T_{ACC}, and A_{DIF} when EVD frequency T_E is changed from 1 to 70 for GEVD- and SEVD-MUSIC. ω_L and ω_H are fixed to 125 and 4,000 Hz, respectively.

It is observed that GEVD and SEVD have the same performance. This is aused by a high signal-to-noise ratio (SNR) of audio data included in tasks 1 and 3. We presume that GEVD will outperform when SNR is lower.

L_{ACC} is maintained even with large T_E, but T_{ACC} and A_{DIF} get worse. This means that T_E affects the detection of sound events less. However, when we carefully look at each pair of sound events, that is, the detected sound event and the corresponding sound event included in ground truth, it can be seen that their correspondence gets poorer as T_E becomes larger. When we go for sound source separation and automatic speech recognition, we should consider a trade-off between fast processing achieved by large T_E and high localization performance by small T_E.

2.2. Comparison of analyzed frequency ranges

The frequency ranges, ω_L and ω_H, were changed in this evaluation. The differences in the three metrics with the results with $(\omega_L; \omega_H) = (125; 4,000)$ were investigated in the following four conditions; $(125; 2,800)$, $(125; 7,500)$, $(500; 2,800)$, and $(500; 4,000)$.

Fig. 4 shows the results. It seems that the performance was not affected by the changes in the analyzed frequency range when it comes to L_{ACC}. However, the results with higher frequencies such as $(500; 2,800)$ and $(500; 4,000)$ showed different performances when we look at T_{ACC} and A_{DIF}. Because the results with almost full frequency range $(125; 7,500)$ had similar performance to lower

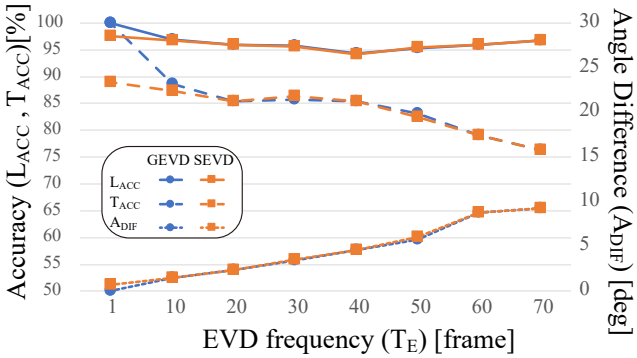

Figure 3: Performance comparison of GEVD- and SEVD-MUSIC: The averaged scores over all files in tasks 1 and 3 were measured with $(\omega_L; \omega_H) = (125; 4,000)$.

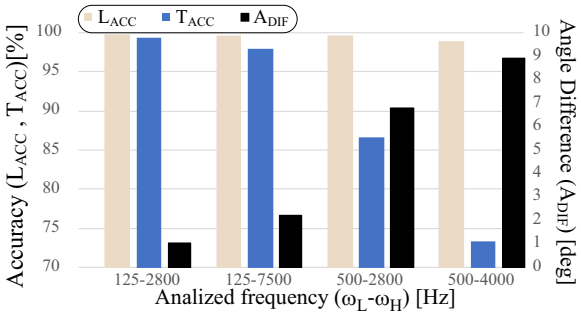

Figure 4: Performance comparison for different frequency ranges: The averaged scores over all files in tasks 1 and 3 were calculated using GEVD with T_E = 1.

frequency ranges such as $(125; 2,800)$ and $(125; 4,000)$, we can say that most of the target signal energy is concentrated on lower frequencies in tasks 1 and 3. With a narrow frequency range, fast processing can be achieved. On the other hand, the frequency range should be matched with that of the target signal. This is another trade-off in sound source localization.

2.3. Comparison for tasks and microphone arrays

Performance differences for all combinations of two tasks and two microphone arrays were measured. For tasks, we selected tasks 1 and 3. Task 1 includes audio files for a single, static loudspeaker using static microphones arrays, and task 3 includes a single, moving talker using static microphone arrays. For microphone arrays. we selected a 12-channel pseudo-spherical array marked as “robot head” and a 32-channel spherical array marked as “eigenmike.”

Fig. 5 shows the results. As mentioned before, SNRs in both tasks are high, and L_{ACC} did not provide significant differences between tasks and between microphone arrays. However, we observed that the performance dropped as EVD performed less frequently from T_{ACC} and A_{DIF}. When T_E is less than 30, there was no difference between the four combinations. However, when T_E is more than 30, it can be seen that there is a remarkable difference between “robot head” and “eigenmike.” The results with “eigenmike” were quite stable. We can say that this is caused by the fact that “eigenmike” consists of more microphones. On the other hand,

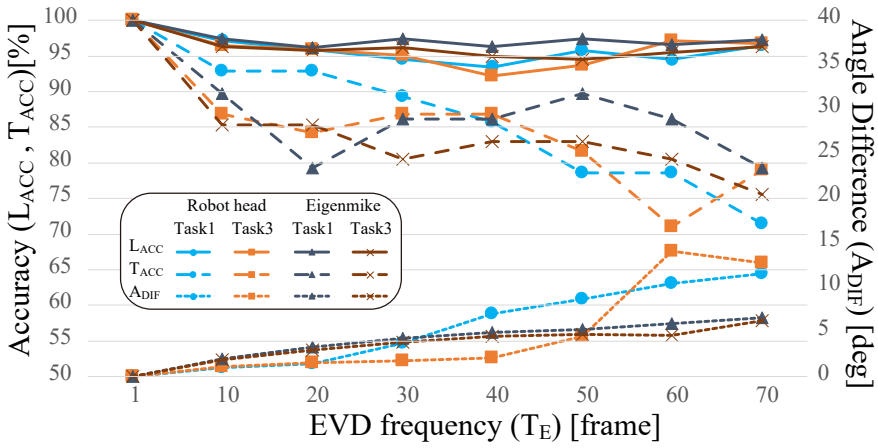

Figure 5: Performance comparison for different microphone arrays in tasks 1 and 3: The averaged scores for each combination of task and microphone array were measured. The frequency range was fixed to (125, 4,000) using GEVD.

there is no difference between tasks 1 and 3. We guess that this is because sound source motions in the task were not rapid.

The acceptable performance will be when A_{DIF} is less than 5 degrees, because the resolution of the steering vectors is 5 degrees. In this sense, T_E should be less than 20. For the processing speed, "robot head" could be processed in real time when T_E is more than 20. For "eigenmike," real-time processing was difficult even with T_E = 70. The best solution for "robot head" is that T_E = 20 when both performance and processing speed are taken into account.

3. SUMMARY

We presented localization results with an offline system consisting of GEVD-MUSIC implemented in HARK, zero-cross based VAD, and sound source tracking with Kalman filtering. For the challenge, we selected four combinations of task 1 and task 3 with a 32-channel spherical array and a 12-channel pseudo-spherical array. The localization results were discussed along with three proposed metrics.

We did not use development data sets, but there is a possibility to improve the localization performance using steering vectors adapted from the development data sets. Deep learning methods can be also another possibility to improve performance [4].

4. ACKNOWLEDGMENT

We thank M. Shimizu, Honda Research Institute Japan for his help. This work was partially supported by JSPS KAKENHI Grant No.16H02884, 16K00294, and 17K00365, and also by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

5. REFERENCES

[1] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[2] T. Ohata, K. Nakamura, A. Nagamine, T. Mizumoto, T. Ishizaki, R. Kojima, O. Sugiyama, and K. Nakadai, "Outdoor sound source detection using a quadcopter with microphone array," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 177–187, 2017.

[3] K. Nakadai, H. G. Okuno, , and T. Mizumoto, "Development, deployment and applications of robot audition open source software hark," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 16–25, 2017.

[4] N. Yalta, K. Nakadai, , and T. Ogata, "Sound source localization using deep learning models," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 37–48, 2017.